Solaris 10 10/08 Installationshandbuch: Benutzerdefinierte JumpStart-Installation und komplexe Installationsszenarien

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Teilenr.: 820–6073–10
Oktober·2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Alle Rechte vorbehalten.

Sun Microsystems, Inc. hat Rechte in Bezug auf geistiges Eigentum an der Technologie, die in dem in diesem Dokument beschriebenen Produkt enthalten ist. Im Besonderen und ohne Einschränkung umfassen diese Ansprüche in Bezug auf geistiges Eigentum eines oder mehrere Patente und eines oder mehrere Patente oder Anwendungen mit laufendem Patent in den USA und in anderen Ländern.

Rechte der US-Regierung – Kommerzielle Software. Regierungsbenutzer unterliegen der standardmäßigen Lizenzvereinbarung von Sun Microsystems, Inc. sowie den anwendbaren Bestimmungen der FAR und ihrer Zusätze.

Diese Ausgabe kann von Drittanbietern entwickelte Bestandteile enthalten.

Teile des Produkts können aus Berkeley BSD-Systemen stammen, die von der University of California lizenziert sind. UNIX ist ein eingetragenes Warenzeichen in den USA und in anderen Ländern und exklusiv durch X/Open Company, Ltd. lizenziert.

Sun, Sun Microsystems, das Sun-Logo, das Solaris-Logo, das Java Kaffeetassenlogo, docs.sun.com, Java und Solaris sind Marken oder eingetragene Marken von Sun Microsystems, Inc. in den USA und anderen Ländern. Alle SPARC-Marken werden unter Lizenz verwendet und sind in den USA und anderen Ländern Marken oder eingetragene Marken von SPARC International, Inc. Produkte, die das SPARC-Markenzeichen tragen, basieren auf einer von Sun Microsystems Inc., entwickelten Architektur.

Die grafischen Benutzeroberflächen von OPEN LOOK und Sun™ wurden von Sun Microsystems, Inc. für seine Benutzer und Lizenznehmer entwickelt. Sun erkennt hiermit die bahnbrechenden Leistungen von Xerox bei der Erforschung und Entwicklung des Konzepts der visuellen und grafischen Benutzeroberfläche für die Computerindustrie an. Sun ist Inhaber einer nicht ausschließlichen Lizenz von Xerox für die grafische Benutzeroberfläche von Xerox. Diese Lizenz gilt auch für Suns Lizenznehmer, die mit den OPEN LOOK-Spezifikationen übereinstimmende Benutzerschnittstellen implementieren und sich an die schriftlichen Lizenzvereinbarungen mit Sun halten.

Produkte, die in dieser Veröffentlichung beschrieben sind, und die in diesem Handbuch enthaltenen Informationen unterliegen den Gesetzen der US-Exportkontrolle und können den Export- oder Importgesetzen anderer Länder unterliegen.. Die Verwendung im Zusammenhang mit Nuklear-, Raketen-, chemischen und biologischen Waffen, im nuklear-maritimen Bereich oder durch in diesem Bereich tätige Endbenutzer, direkt oder indirekt, ist strengstens untersagt. Der Export oder Rückexport in Länder, die einem US-Embargo unterliegen, oder an Personen und Körperschaften, die auf der US-Exportausschlussliste stehen, einschließlich (jedoch nicht beschränkt auf) der Liste nicht zulässiger Personen und speziell ausgewiesener Staatsangehöriger, ist strengstens untersagt.

DIE DOKUMENTATION WIRD “AS IS” BEREITGESTELLT, UND JEGLICHE AUSDRÜCKLICHE ODER IMPLIZITE BEDINGUNGEN, DARSTELLUNGEN UND HAFTUNG, EINSCHLIESSLICH JEGLICHER STILLSCHWEIGENDER HAFTUNG FÜR MARKTFÄHIGKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ODER NICHTÜBERTRETUNG WERDEN IM GESETZLICH ZULÄSSIGEN RAHMEN AUSDRÜCKLICH AUSGESCHLOSSEN.

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

081114@21288

Inhalt

Vorwort 9

Teil I Arbeiten mit der benutzerdefinierten JumpStart-Installation 13

1 Informationen zur Planung einer Solaris-Installation 15
Informationen zur Planung und zu den Systemanforderungen 15

2 Benutzerdefinierte JumpStart-Installation (Übersicht) 17
Einführung in die benutzerdefinierte JumpStart-Installation 17
Beispielszenario für die benutzerdefinierte JumpStart-Installation 18
So installiert das JumpStart-Programm die Solaris-Software 19

3 Vorbereiten von benutzerdefinierten JumpStart-Installationen (Vorgehen) 23
Übersicht der Schritte: Vorbereiten von benutzerdefinierten JumpStart-Installationen 24
Erstellen eines Profilservers für vernetzte Systeme 25
▼ So erstellen Sie ein JumpStart-Verzeichnis auf einem Server 26
Allen Systemen Zugriff auf den Profilserver gewähren 27
Erstellen einer Profildiskette für Standalone-Systeme 30
▼ SPARC: So erstellen Sie eine Profildiskette 30
▼ x86: So erstellen Sie eine Profildiskette mit GRUB 32
Erstellen der Datei `rules` 34
Syntax der Datei `rules` 35
▼ So erstellen Sie die Datei `rules` 35
`rules`-Beispieldatei 37
Erstellen eines Profils 38
Syntax von Profilen 39
▼ So erstellen Sie ein Profil 39

Beispiele für Profile 40
Testen eines Profils 52
▼ So erstellen Sie eine temporäre Solaris-Umgebung zum Testen eines Profils 53
▼ So testen Sie ein Profil 54
Beispiele für das Testen von Profilen 56
Validieren der Datei `rules` 57
▼ So validieren Sie die Datei `rules` 58

4 Verwenden der optionalen Funktionen der benutzerdefinierten JumpStart-Installation (Vorgehen) 59
Erstellen von Begin-Skripten 59
Wichtige Informationen zu Begin-Skripten 60
Erstellen abgeleiteter Profile mit einem Begin-Skript 60
Erstellen von Finish-Skripten 62
Wichtige Informationen zu Finish-Skripten 62
▼ So fügen Sie Dateien mit einem Finish-Skript hinzu 62
Hinzufügen von Packages oder Patches mit einem Finish-Skript 63
Anpassen der Root-Umgebung mit einem Finish-Skript 66
Definieren des Root-Passworts eines Systems mit einem Finish-Skript 66
Automatische Installationen mit Finish-Skripten 68
Erstellen einer komprimierten Konfigurationsdatei 69
▼ So erstellen Sie eine komprimierte Konfigurationsdatei 69
Beispiel für eine komprimierte Konfigurationsdatei 70
Erstellen von Festplattenkonfigurationsdateien 70
▼ SPARC: So erstellen Sie eine Festplattenkonfigurationsdatei 70
SPARC: Beispiel für eine Festplattenkonfigurationsdatei 71
▼ x86: So erstellen Sie eine Festplattenkonfigurationsdatei 72
x86: Beispiel für eine Festplattenkonfigurationsdatei 74
Verwenden eines standortspezifischen Installationsprogramms 76

5 Erstellen von benutzerdefinierten Rule- und Probe-Schlüsselwörtern (Vorgehen) 77
Probe-Schlüsselwörter 77
Erstellen einer `custom_probes`-Datei 78
Syntax der Datei `custom_probes` 78
Syntax von Funktionsnamen in `custom_probes` 79

▼ So erstellen Sie eine `custom_probes`-Datei 79
Beispiele für `custom_probes`-Dateien und Schlüsselwörter 80
Validieren der Datei `custom_probes` 81
▼ So validieren Sie die Datei `custom_probes` 81

6 Ausführen einer benutzerdefinierten JumpStart-Installation (Vorgehen) 83
Einschränkungen für eine JumpStart-Installation 83
SPARC: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation 85
SPARC: Ausführen einer benutzerdefinierten JumpStart-Installation 86
▼ So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor 87
▼ SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus 89
SPARC: Befehlsreferenz für den Befehl `boot` 90
x86: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation 92
x86: Ausführen einer benutzerdefinierten JumpStart-Installation 93
▼ x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus 94
x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB 96
x86: Befehlsreferenz zum Booten des Systems 98

7 Benutzerdefinierte JumpStart-Installation (Beispiele) 101
Konfiguration des Beispielstandorts 102
Erstellen eines Installationsservers 103
x86: Erstellen eines Boot-Servers für die Marketingsysteme 104
Erstellen eines JumpStart-Verzeichnisses 105
Freigeben des JumpStart-Verzeichnisses zur gemeinsamen Nutzung 105
SPARC: Erstellen des Profils für die Entwicklungsgruppe 106
x86: Erstellen des Profils für die Marketinggruppe 106
Aktualisieren der Datei `rules` 107
Validieren der Datei `rules` 108
SPARC: Einrichten der Entwicklungssysteme für die Installation über das Netzwerk 108
x86: Einrichten der Marketingsysteme für die Installation über das Netzwerk 109

SPARC: Booten der Entwicklungssysteme und Installation der Solaris-Software 110
x86: Booten der Marketingsysteme und Installation der Solaris-Software 110

8 Benutzerdefinierte JumpStart-Installation (Referenz) 111
Rule-Schlüsselwörter und -Werte 111
Profilschlüsselwörter und -werte 116
Profilschlüsselwörter im Überblick 116
Profilschlüsselwörter - Beschreibung und Beispiele 118
Benutzerdefinierte JumpStart-Umgebungsvariablen 166
Probe-Schlüsselwörter und -Werte 168

9 Installation eines ZFS-Root-Pools mit JumpStart 171
JumpStart-Installation für ein ZFS-Root-Dateisystem (/) (Übersicht und Planung) 171
Einschränkungen für eine JumpStart-Installation für ein ZFS-Root-Pool 172
JumpStart-Profilbeispiele für ein ZFS-Root-Pool 173
JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick) 177
`bootenv`-Profilschlüsselwort (ZFS und UFS) 178
`install_type`-Schlüsselwort (ZFS und UFS) 179
`pool`-Profilschlüsselwort (nur ZFS) 179
`root_device`-Profilschlüsselwort (ZFS und UFS) 181
Weitere Ressourcen 182

Teil II Anhänge 183

A Fehlerbehebung (Vorgehen) 185
Probleme beim Einrichten von Netzwerkinstallationen 185
Probleme beim Booten eines Systems 186
Booten von Medien, Fehlermeldungen 186
Booten von Medien, allgemeine Probleme 187
Booten vom Netzwerk, Fehlermeldungen 188
Booten vom Netzwerk, allgemeine Probleme 191
Neuinstallation von Solaris 192
▼ x86: So überprüfen Sie eine IDE-Festplatte auf fehlerhafte Blöcke 193
Upgrade von Solaris 194

Durchführen eines Upgrade, Fehlermeldungen 194
Durchführen eines Upgrade, allgemeine Probleme 196
▼ So setzen Sie ein Upgrade nach einem Fehlschlag fort 198
x86: Probleme mit Solaris Live Upgrade bei der Verwendung von GRUB 198
▼ Systempanik bei einem Upgrade mit Solaris Live Upgrade und Veritas VxVm 200
x86: Service-Partition wird auf Systemen ohne bereits vorhandene Service-Partition nicht standardmäßig erzeugt 203
▼ So installieren Sie die Software von einem Netzwerk-Installationsabbild oder der Solaris Operating System-DVD 203
▼ So installieren Sie von der Solaris Software-1-CD oder einem Netzwerk-Installationsabbild 204

B Zusätzliche SVR4-Packaging-Anforderungen (Referenz) 205
Verhindern einer Modifikation des aktuellen BS 205
Verwenden absoluter Pfade 205
Verwenden des Befehls `pkgadd -R` 206
Unterschiede zwischen `$PKG_INSTALL_ROOT` und `$BASEDIR` - Übersicht 207
Richtlinien zum Schreiben von Skripten 207
Erhalten der Diskless-Client-Kompatibilität 208
Überprüfen von Packages 208
Verhindern der Benutzerinteraktion bei Installation oder Upgrade 209
Einstellen von Package-Parametern für Zonen 211
Hintergrundinformationen 214

Glossar 215

Index 231

Vorwort

Dieses Dokument beschreibt die Installation und das Upgrade des Betriebssystems (BS) Solaris™ auf vernetzten und nicht vernetzten SPARC®- und x86-basierten Systemen. Dabei werden auch die benutzerdefinierte JumpStart-Installation sowie die Erstellung von RAID-1-Volumes bei der Installation behandelt.

Dieses Handbuch enthält keine Informationen zum Konfigurieren von Systemhardware und Peripheriegeräten.

Hinweis – Dieses Solaris-Release unterstützt Systeme auf der Basis der Prozessorarchitekturen SPARC und x86: UltraSPARC®, SPARC64, AMD64, Pentium und Xeon EM64T. Die unterstützten Systeme können Sie in der *Solaris OS: Hardware-Kompatibilitätsliste* unter `http://www.sun.com/bigadmin/hcl` nachlesen. Eventuelle Implementierungsunterschiede zwischen den Plattformtypen sind in diesem Dokument angegeben.

In diesem Dokument bedeuten x86-bezogene Begriffe Folgendes:

- „x86" bezeichnet die weitere Familie an Produkten, die mit 64-Bit- und 32-Bit-x86-Architekturen kompatibel sind.
- „x64" weist auf spezifische, für 64-Bit-Systeme geltende Informationen zu AMD64- bzw. EM64T-Systemen hin.
- „32-Bit x86" weist auf spezifische, für 32-Bit-Systeme geltende Informationen zu x86-basierten Systemen hin.

Die unterstützten Systeme können Sie der *Solaris OS: Hardware-Kompatibilitätsliste* entnehmen.

Zielgruppe dieses Handbuchs

Dieses Handbuch richtet sich an Systemadministratoren, die für die Installation des Betriebssystems Solaris zuständig sind. Das Handbuch enthält folgende Informationen:

- Weiterführende Informationen zur Installation von Solaris für Systemadministratoren in Unternehmen, die mehrere Solaris-Rechner in einer vernetzten Umgebung verwalten
- Grundlegende Informationen zur Installation von Solaris für Systemadministratoren, die Solaris-Installationen bzw. -Upgrades nur von Zeit zu Zeit vornehmen

Verwandte Dokumentation

In Tabelle P–1 ist die Dokumentation für Systemadministratoren aufgeführt.

TABELLE P–1 Sind Sie ein Systemadministrator, der Solaris installiert?

Beschreibung	Informationen
Benötigen Sie die Systemvoraussetzungen oder Informationen zur allgemeinen Planung? Benötigen Sie eine allgemeine Übersicht zu Solaris ZFS™-Installationen, zum Booten, zur Partitionierungstechnologie Solaris Zones oder zum Erstellen von RAID-1-Volumes?	*Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
Müssen Sie ein System von einer DVD oder CD installieren? Das Solaris-Installationsprogramm führt Sie Schritt für Schritt durch die Installation.	*Solaris 10 10/08 Installationshandbuch: Grundinstallationen*
Müssen Sie Ihr System patchen oder aktualisieren, und darf es dabei möglichst nicht zu einer Ausfallzeit kommen? Aktualisieren Sie Ihr System mit Solaris Live Upgrade, um die Ausfallzeit auf ein Minimum zu reduzieren.	*Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*
Müssen Sie eine sichere Installation über das Netzwerk oder das Internet durchführen? Verwenden Sie WAN-Boot, um auf einem remoten Client zu installieren. Oder müssen Sie über das Netzwerk von einem Installationsabbild installieren? Das Solaris-Installationsprogramm führt Sie Schritt für Schritt durch die Installation.	*Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
Müssen Sie mehrere Systeme schnell aktualisieren oder patchen? Verwenden Sie die Solaris Flash™-Software, um ein Solaris Flash-Archiv zu erstellen und eine Kopie des Betriebssystems auf Klonsystemen zu installieren.	*Solaris 10 10/08 Installationshandbuch: Solaris Flash-Archive (Erstellung und Installation)*
Müssen Sie Ihr System sichern?	Kapitel 23, „Backing Up and Restoring UFS File Systems (Overview)“ in *System Administration Guide: Devices and File Systems*
Benötigen Sie Informationen zur Fehlerbehebung, eine Liste der bekannten Probleme oder eine Liste der Patches für diese Version?	*Solaris Versionshinweise*
Müssen Sie überprüfen, ob Ihr System für die Ausführung von Solaris geeignet ist?	SPARC: *Solaris Handbuch zur Hardware-Plattform von Sun*
Müssen Sie überprüfen, welche Pakete in dieser Version hinzugefügt, entfernt oder geändert wurden?	*Solaris Package List*
Müssen Sie überprüfen, ob Ihr System und Ihre Geräte mit Solaris SPARC- und x86-basierten Systemen und anderen Drittanbietern ausgeführt werden können?	Solaris Hardware Compatibility List for x86 Platforms

Dokumentation, Support und Schulung

Auf der Sun-Website finden Sie Informationen zu den folgenden zusätzlichen Ressourcen:

- Dokumentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Schulung (`http://www.sun.com/training/`)

Typografische Konventionen

In der folgenden Tabelle sind die in diesem Handbuch verwendeten typografischen Konventionen aufgeführt.

TABELLE P–2 Typografische Konventionen

Schriftart	Bedeutung	Beispiel
`AaBbCc123`	Die Namen von Befehlen, Dateien, Verzeichnissen sowie Bildschirmausgabe.	Bearbeiten Sie Ihre `.login`-Datei. Verwenden Sie `ls -a`, um eine Liste aller Dateien zu erhalten. `system% Sie haben eine neue Nachricht.`
`AaBbCc123`	Von Ihnen eingegebene Zeichen (im Gegensatz zu auf dem Bildschirm angezeigten Zeichen)	`system%` **`su`** `Passwort:`
aabbcc123	Platzhalter: durch einen tatsächlichen Namen oder Wert zu ersetzen	Geben Sie zum Löschen einer Datei den Befehl `rm` *Dateiname* ein.
AaBbCc123	Buchtitel, neue Ausdrücke; hervorgehobene Begriffe	Lesen Sie hierzu Kapitel 6 im *Benutzerhandbuch*. Ein *Cache* ist eine lokal gespeicherte Kopie. Diese Datei *nicht* speichern. **Hinweis:** Einige hervorgehobene Begriffe werden online fett dargestellt.

Shell-Eingabeaufforderungen in Befehlsbeispielen

Die folgende Tabelle zeigt die Standard-Systemeingabeaufforderung von UNIX® und die Superuser-Eingabeaufforderung für die C-Shell, die Bourne-Shell und die Korn-Shell.

TABELLE P–3 Shell-Eingabeaufforderungen

Shell	Eingabeaufforderung
C-Shell	`system%`
C-Shell für Superuser	`system#`
Bourne-Shell und Korn-Shell	`$`
Bourne-Shell und Korn-Shell für Superuser	`#`

TEIL I

Arbeiten mit der benutzerdefinierten JumpStart-Installation

In diesem Teil finden Sie Anweisungen zum Anlegen, Vorbereiten und Ausführen einer benutzerdefinierten JumpStart-Installation.

◆ ◆ ◆ KAPITEL 1

Informationen zur Planung einer Solaris-Installation

Dieses Buch enthält Informationen zur Verwendung des automatisierten JumpStart-Installationsprogramms zur Installation des Betriebssystems Solaris. Es stellt Ihnen alle erforderlichen Informationen zur Installation mit dem JumpStart-Programm zur Verfügung. Dennoch empfehlen wir Ihnen die Lektüre eines Planungshandbuches aus der Installations-Dokumentationsreihe, bevor Sie mit den Vorbereitungen für eine JumpStart-Installation beginnen. Die folgende Referenz enthält nützliche Informationen, die Sie vor der Installation Ihres Systems lesen sollten.

Informationen zur Planung und zu den Systemanforderungen

Im *Solaris 10 10/08Installationshandbuch: Planung von Installationen und Upgrades* sind die Systemanforderungen und allgemeine Informationen zur Planung aufgeführt, beispielsweise Planungsrichtlinien für Dateisysteme, Planung von Upgrades und vieles weiteres mehr. Dieser Abschnitt enthält eine Übersicht der Kapitel in diesem Handbuch.

Beschreibung der Kapitel im Planungshandbuch	Referenz
In diesem Kapitel werden neue Funktionen in den Solaris-Installationsprogrammen beschrieben.	Kapitel 2, „Neuerungen in der Solaris-Installation“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
In diesem Kapitel finden Sie Informationen zu den Entscheidungen, die Sie treffen müssen, bevor Sie das Betriebssystem Solaris installieren oder ein Upgrade ausführen. Beispiele zur Entscheidungsfindung, ob ein Installationsimage aus dem Netzwerk oder eine DVD zur Installation verwendet werden soll.	Kapitel 3, „Installation und Upgrade von Solaris (Roadmap)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*

Beschreibung der Kapitel im Planungshandbuch	Referenz
Dieses Kapitel befasst sich mit den Systemvoraussetzungen für eine Installation oder ein Upgrade des Betriebssystems (BS) Solaris. Außerdem enthält es allgemeine Richtlinien für die Planung der Zuordnung von Speicherplatz und Standard-Swap-Platz. Hier finden Sie auch Informationen zu den Einschränkungen bei Upgrades.	Kapitel 4, „Systemvoraussetzungen, Richtlinien und Upgrades (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
Dieses Kapitel enthält Checklisten, mit deren Hilfe Sie die Informationen zusammenstellen können, die Sie für eine Installation bwz. ein Upgrade benötigen. Diese Informationen sind zum Beispiel beim Ausführen einer interaktiven Installation nützlich. Alle Informationen, die Sie für eine interaktive Installation benötigen, befinden sich in einer Checkliste.	Kapitel 5, „Zusammenstellen von Informationen vor einer Installation bzw. einem Upgrade (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
Diese Kapitel enthalten Übersichten der verschiedenen Technologien, die mit der Installation oder Aktualisierung des Betriebssystems Solaris in Verbindung stehen. Außerdem finden Sie hier Richtlinien und Anforderungen zu diesen Technologien. Diese Kapitel enthalten Informationen zu folgenden Themen: ZFS-Installationen, Booten, Partitionierungstechnologie Solaris Zones und RAID-1-Volumes, die während der Installation angelegt werden können	Teil II, „Installationen in Verbindung mit ZFS, Booten, Solaris Zones und RAID-1-Volumes“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*

◆ ◆ ◆ KAPITEL 2

Benutzerdefinierte JumpStart-Installation (Übersicht)

Dieses Kapitel enthält eine Einführung in und eine Übersicht über das benutzerdefinierte JumpStart-Installationsverfahren.

Hinweis – Wenn Sie ein Solaris-ZFS™-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „Einführung in die benutzerdefinierte JumpStart-Installation“ auf Seite 17
- „So installiert das JumpStart-Programm die Solaris-Software“ auf Seite 19

Einführung in die benutzerdefinierte JumpStart-Installation

Die benutzerdefinierte JumpStart-Installation bietet eine Befehlszeilenschnittstelle, mit der Sie automatisch auf mehreren Systemen eine Installation bzw. ein Upgrade ausführen können, und zwar basierend auf von Ihnen erstellten Profilen. Diese Profile definieren die spezifischen Software-Installationsanforderungen. Außerdem können Sie für die vor und nach der Installation erforderlichen Schritte Shell-Skripte verwenden. Dabei geben Sie selbst an, welche Profile und Skripte für die Installation bzw. das Upgrade verwendet werden sollen. Die Installation bzw. das Upgrade mit der benutzerdefinierten JumpStart-Installation wird dann auf der Grundlage der von Ihnen ausgewählten Profile und Skripte ausgeführt. Außerdem können Sie eine `sysidcfg`-Datei verwenden und die Konfigurationsinformationen vorkonfigurieren, so dass die benutzerdefinierte JumpStart-Installation völlig ohne Benutzereingriff abläuft.

Beispielszenario für die benutzerdefinierte JumpStart-Installation

Die benutzerdefinierte JumpStart-Installation lässt sich anhand des folgenden Beispielszenarios beschreiben. In diesem Beispielszenario müssen die Systeme mit den folgenden Parametern eingerichtet werden:

- Installation von Solaris auf 100 neuen Systemen.
- Bei siebzig dieser Systeme handelt es sich um SPARC-basierte Systeme der technischen Entwicklungsabteilung. Sie müssen als Standalone-Systeme mit der Solaris-Softwaregruppe für Entwickler installiert werden.
- Die übrigen 30 Systeme sind x86-basiert, werden von der Marketingabteilung genutzt und müssen als Standalone-Systeme mit der Solaris-Softwaregruppe für Endbenutzer installiert werden.

Zunächst muss der Systemadministrator eine `rules`-Datei und ein Profil für jede Gruppe von Systemen erstellen. Die Datei `rules` ist eine Textdatei, die eine Regel (rule) für jede Systemgruppe oder ein einzelnes System enthält, auf der bzw. dem Sie die Solaris-Software installieren wollen. Jede Regel charakterisiert eine Gruppe von Systemen auf der Grundlage von einem oder mehreren Systemattributen. Jede Regel verknüpft außerdem jede Gruppe mit einem Profil.

Ein Profil ist eine Textdatei, in der definiert ist, wie die Solaris-Software auf den Systemen in der Gruppe installiert werden soll. Die Datei `rules` und das Profil müssen sich in einem JumpStart-Verzeichnis befinden.

In diesem Beispielszenario erstellt der Systemadministrator eine `rules`-Datei, die zwei Regeln enthält, eine für die Entwicklungs- und eine für die Marketinggruppe. Bei beiden Regeln dient die Netzwerknummer der Systeme zur Unterscheidung zwischen der Entwicklungs- und der Marketinggruppe.

Jede Regel enthält außerdem eine Verknüpfung zu einem entsprechenden Profil. In der Regel für die Entwicklungsgruppe wird zum Beispiel eine Verknüpfung zum Profil `eng_profile` hinzugefügt, das für die Entwicklungsgruppe erstellt wurde. In der Regel für die Marketinggruppe wird eine Verknüpfung zum Profil `market_profile` hinzugefügt, das für die Marketinggruppe erstellt wurde.

Sie können die Datei `rules` und die Profile auf einer Diskette oder einem Server speichern.

- Eine Profildiskette wird benötigt, wenn Sie benutzerdefinierte JumpStart-Installationen auf nicht vernetzten Standalone-Systemen ausführen wollen.
- Ein Profilserver wird benötigt, wenn Sie benutzerdefinierte JumpStart-Installationen auf vernetzten Systemen ausführen wollen, die Zugriff auf einen Server haben.

Nachdem Sie die Datei `rules` und die Profile erstellt haben, validieren Sie die Dateien mit dem Skript `check`. Bei erfolgreicher Ausführung des Skripts `check` wird die Datei `rules.ok` erstellt. Die Datei `rules.ok` ist eine generierte Version der Datei `rules`, die das JumpStart-Programm zur Installation der Solaris-Software verwendet.

So installiert das JumpStart-Programm die Solaris-Software

Nachdem Sie die Datei `rules` und die Profile validiert haben, können Sie eine benutzerdefinierte JumpStart-Installation starten. Das JumpStart-Programm liest die Datei `rules.ok`. Danach sucht das JumpStart-Programm die erste Regel mit definierten Systemattributen, die dem System entsprechen, auf dem die Solaris-Software mit dem JumpStart-Programm gerade installiert werden soll. Wenn eine Entsprechung gefunden wird, verwendet das JumpStart-Programm das in der Regel angegebene Profil zur Installation der Solaris-Software auf dem System.

Abbildung 2–1 zeigt, wie eine benutzerdefinierte JumpStart-Installation auf einem eigenständigen, nicht vernetzten System funktioniert. Der Systemadministrator initiiert die benutzerdefinierte JumpStart-Installation auf Martins System. Das JumpStart-Programm greift auf die Regeldatei auf der Diskette im Diskettenlaufwerk des Systems zu. Das JumpStart-Programm ordnet `Regel 2` dem System zu. `Regel 2` legt fest, dass das JumpStart-Programm `Martins Profil` zur Installation der Solaris-Software verwendet. Das JumpStart-Programm liest `Martins Profil` und installiert die Solaris-Software nach den Anweisungen, die der Systemadministrator in `Martins Profil` festgelegt hat.

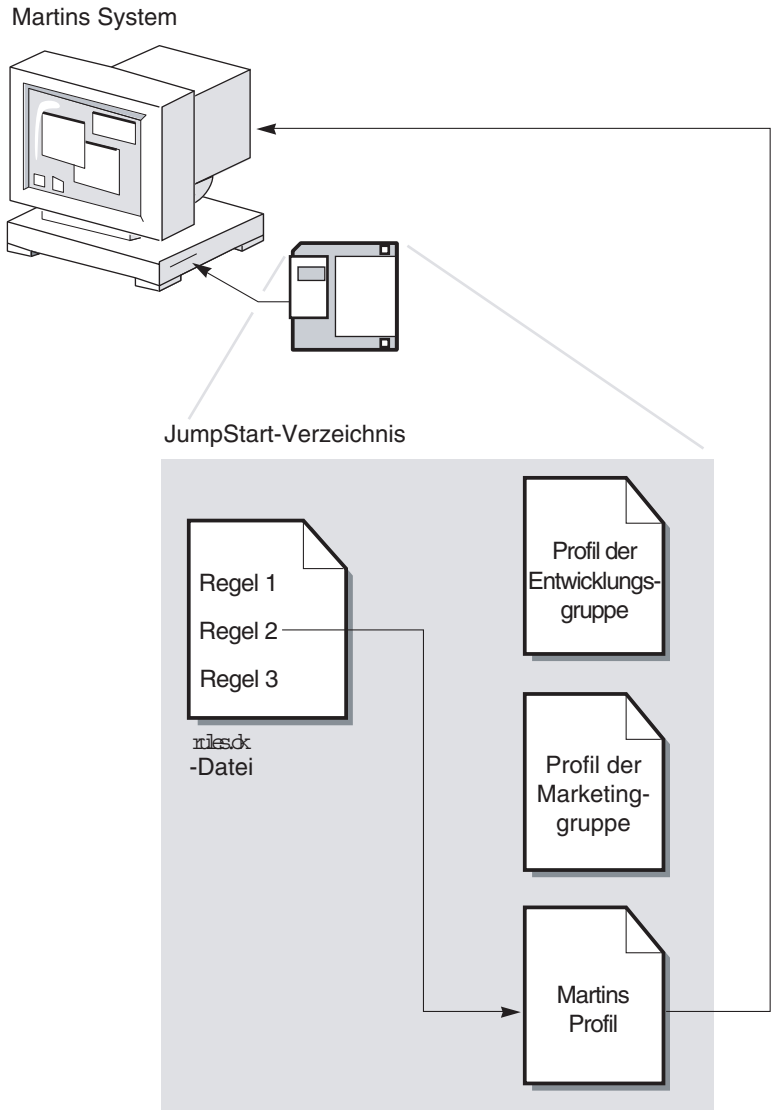

ABBILDUNG 2–1 Prinzip einer benutzerdefinierten JumpStart-Installation: Beispiel für nicht vernetzte Systeme

Abbildung 2–2 zeigt, wie eine benutzerdefinierte JumpStart-Installation bei mehreren Systemen in einem Netzwerk funktioniert. Der Systemadministrator richtet verschiedene Profile ein und speichert diese auf einem Server. Der Systemadministrator initiiert die benutzerdefinierte JumpStart-Installation auf einem der Systeme der Entwicklungsgruppe. Das JumpStart-Programm greift auf die rules-Dateien im Verzeichnis `JumpStart/` auf dem Server zu. Das JumpStart-Programm ordnet das Entwicklungssystem der `Regel 1` zu. `Regel 1` legt

fest, dass das JumpStart-Programm das `Profil der Entwicklungsgruppe` zur Installation der Solaris-Software verwendet. Das JumpStart-Programm liest das `Profil der Entwicklungsgruppe` und installiert die Solaris-Software nach den Anweisungen, die der Systemadministrator im `Profil der Entwicklungsgruppe` festgelegt hat.

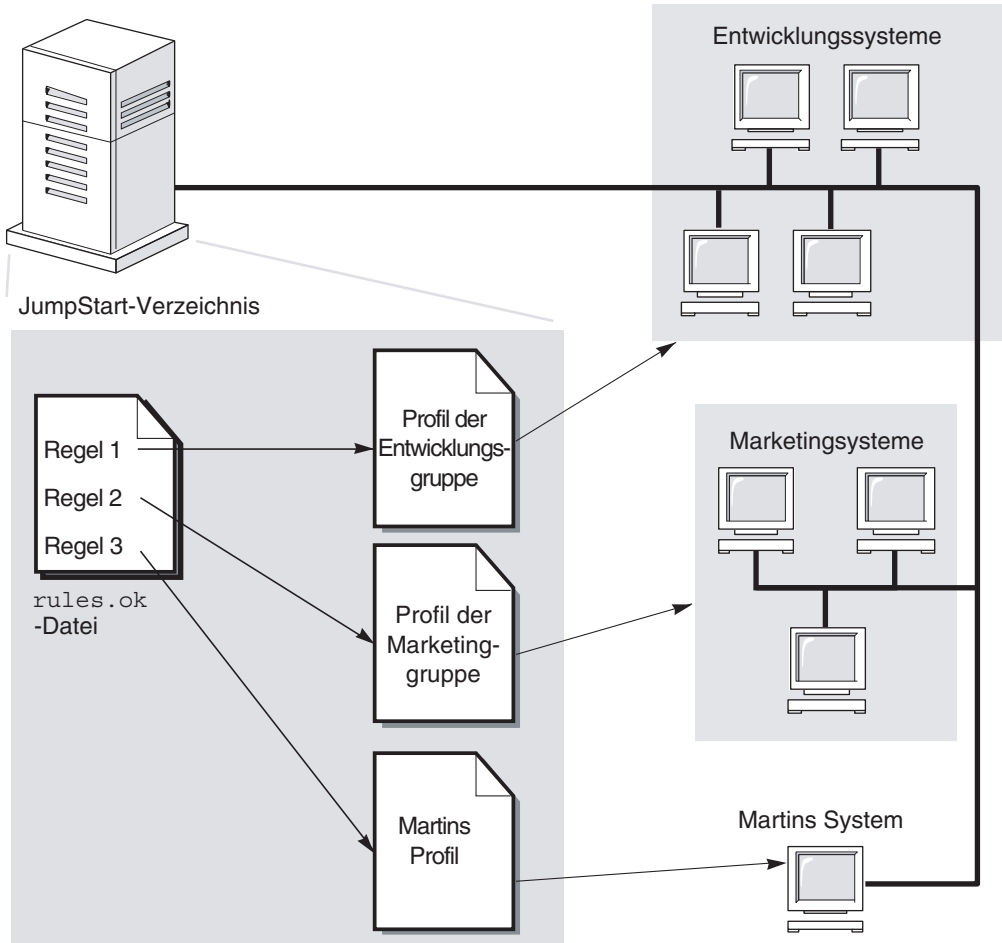

ABBILDUNG 2–2 Prinzip einer benutzerdefinierten JumpStart-Installation: Beispiel für vernetzte Systeme

Abbildung 2–3 beschreibt die Reihenfolge, in der das JumpStart-Programm nach den benutzerdefinierten JumpStart-Dateien sucht.

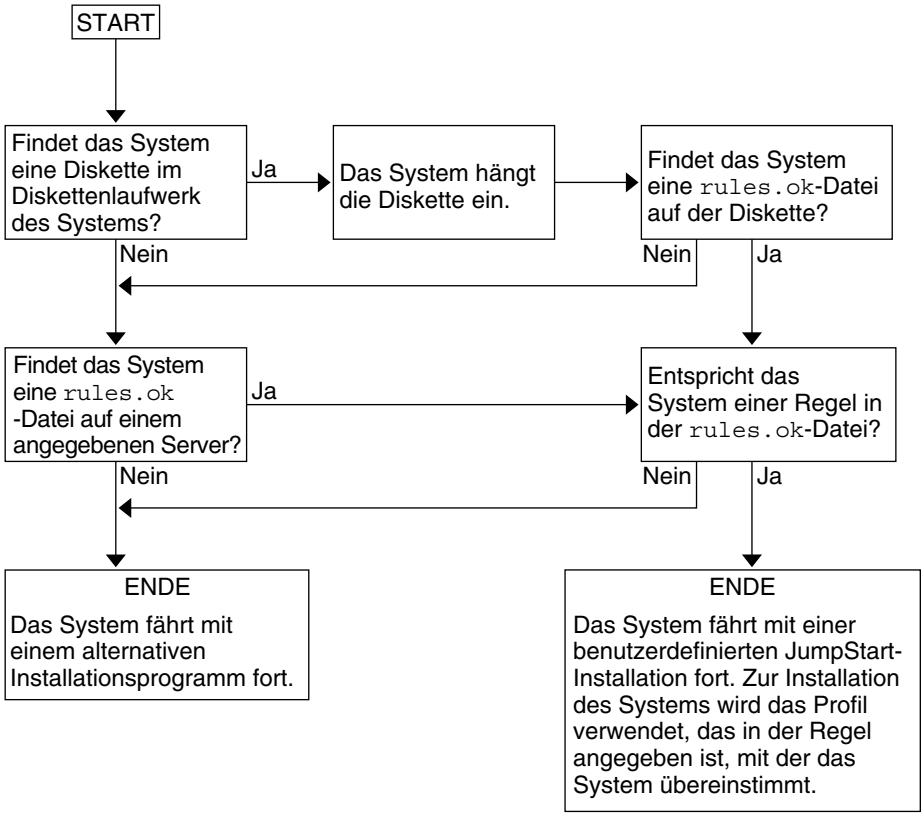

ABBILDUNG 2–3 Ablauf einer benutzerdefinierten JumpStart-Installation

◆ ◆ ◆ KAPITEL 3

Vorbereiten von benutzerdefinierten JumpStart-Installationen (Vorgehen)

In diesem Kapitel wird Schritt für Schritt erläutert, wie Sie die Systeme vorbereiten, von denen und auf denen Sie die Solaris-Software mithilfe des benutzerdefinierten JumpStart-Installationsverfahrens installieren wollen.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „Übersicht der Schritte: Vorbereiten von benutzerdefinierten JumpStart-Installationen“ auf Seite 24
- „Erstellen eines Profilservers für vernetzte Systeme“ auf Seite 25
- „Erstellen einer Profildiskette für Standalone-Systeme“ auf Seite 30
- „Erstellen der Datei `rules`“ auf Seite 34
- „Erstellen eines Profils“ auf Seite 38
- „Testen eines Profils“ auf Seite 52
- „Validieren der Datei `rules`“ auf Seite 57

Übersicht der Schritte: Vorbereiten von benutzerdefinierten JumpStart-Installationen

TABELLE 3–1 Übersicht der Schritte: Vorbereiten von benutzerdefinierten JumpStart-Installationen

Aufgabe	Beschreibung	Siehe
Entscheiden Sie, wie ein Upgrade des Systems ausgeführt werden soll, wenn eine frühere Version der Solaris-Software bereits auf dem System installiert ist.	Wenn bereits ein früheres Solaris-Release auf dem System installiert ist, müssen Sie entscheiden, wie das Upgrade ausgeführt werden soll. Stellen Sie sicher, dass Sie wissen, was vor und nach einem Upgrade eines Systems zu tun ist. Eine sorgfältige Planung erleichtert das Erstellen von Profilen, Begin- und Finish-Skripten.	„Planung von Upgrades“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
Erstellen Sie ein JumpStart-Verzeichnis.	**Auf einem Server** Wenn Sie benutzerdefinierte JumpStart-Installationen auf Systemen ausführen wollen, die an ein Netzwerk angeschlossen sind, müssen Sie einen Profilserver erstellen. Der Profilserver enthält ein JumpStart-Verzeichnis für die benutzerdefinierten JumpStart-Dateien. **Auf einer Diskette** Wenn Sie benutzerdefinierte JumpStart-Installationen auf Systemen ausführen wollen, die nicht an ein Netzwerk angeschlossen sind, müssen Sie eine Profildiskette erstellen. Die Profildiskette enthält die benutzerdefinierten JumpStart-Dateien.	„Erstellen eines Profilservers für vernetzte Systeme“ auf Seite 25 „Erstellen einer Profildiskette für Standalone-Systeme“ auf Seite 30
Fügen Sie Regeln zu der Datei `rules` hinzu.	Nachdem Sie festgelegt haben, wie die einzelnen Systemgruppen oder Einzelsysteme installiert werden sollen, erstellen Sie für jede Gruppe eine Regel. Jede Regel charakterisiert eine Gruppe auf der Grundlage von einem oder mehreren Systemattributen. Die Regel verknüpft die jeweilige Gruppe mit einem Profil.	„Erstellen der Datei `rules`“ auf Seite 34

TABELLE 3–1 Übersicht der Schritte: Vorbereiten von benutzerdefinierten JumpStart-Installationen *(Fortsetzung)*

Aufgabe	Beschreibung	Siehe
Erstellen Sie für jede Regel ein Profil.	Ein Profil ist eine Textdatei, in der definiert ist, wie die Solaris-Software installiert werden soll, also zum Beispiel welche Softwaregruppe auf einem System installiert werden soll. Jede Regel gibt ein Profil an, das definiert, wie ein System installiert werden soll, wenn es der Regel entspricht. Sie können für jede Regel ein eigenes Profil erstellen. Sie können ein Profil jedoch auch in mehreren Regeln verwenden.	„Erstellen eines Profils“ auf Seite 38
(Optional) Testen Sie die Profile.	Nachdem Sie ein Profil erstellt haben, testen Sie das Profil mit dem Befehl `pfinstall(1M)`, bevor Sie es zur Installation bzw. zu einem Upgrade eines Systems verwenden.	„Testen eines Profils“ auf Seite 52
Validieren Sie die Datei `rules`.	Die Datei `rules.ok` ist eine generierte Version der Datei `rules`, anhand derer das JumpStart-Programm ermittelt, welches Profil bei der Installation eines Systems verwendet werden soll. Zum Validieren der Datei `rules` müssen Sie das Skript `check` verwenden.	„Validieren der Datei `rules`“ auf Seite 57

Erstellen eines Profilservers für vernetzte Systeme

Beim Einrichten von benutzerdefinierten JumpStart-Installationen für Systeme im Netzwerk müssen Sie ein Verzeichnis auf einem Server erstellen, das als JumpStart-Verzeichnis bezeichnet wird. Das JumpStart-Verzeichnis enthält alle wichtigen benutzerdefinierten JumpStart-Dateien, wie zum Beispiel die Datei `rules`, die Datei `rules.ok` und die Profile. Das JumpStart-Verzeichnis muss im Root-Verzeichnis (`/`) des Profilservers angelegt werden.

Der Server, auf dem sich das JumpStart-Verzeichnis befindet, wird als Profilserver bezeichnet. Sie können den Installations- oder Boot-Server als Profilserver einrichten oder einen ganz anderen Server verwenden. Ein Profilserver kann JumpStart-Dateien für unterschiedliche Plattformen enthalten. So können Sie zum Beispiel auf einem x86-Server benutzerdefinierte JumpStart-Dateien für SPARC- und x86-basierte Systeme bereitstellen.

Hinweis – Nachdem Sie einen Profilserver erstellt haben, müssen Sie den Systemen Zugriff auf den Server gewähren. Weitere Informationen finden Sie unter „So gewähren Sie den Systemen Zugriff auf den Profilserver“ auf Seite 28.

▼ So erstellen Sie ein JumpStart-Verzeichnis auf einem Server

Hinweis – Bei diesem Verfahren wird davon ausgegangen, dass *Volume Manager* auf dem System läuft. Wenn Sie zum Verwalten von Datenträgern nicht Volume Manager verwenden, finden Sie Näheres zum Verwalten von Wechseldatenträgern ohne Volume Manager im *System Administration Guide: Devices and File Systems* .

1. **Suchen Sie den Server, auf dem Sie das JumpStart-Verzeichnis erstellen wollen.**

2. **Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.**

   Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

3. **Erstellen Sie das JumpStart-Verzeichnis an einer beliebigen Stelle auf dem Server.**

   `# mkdir -m 755` *Jumpstart-Verz_pfad*

   In diesem Befehl ist *Jumpstart-Verz_pfad* der absolute Pfad zum JumpStart-Verzeichnis.

   Mit dem folgenden Befehl wird zum Beispiel ein Verzeichnis mit dem Namen `jumpstart` im Root-Verzeichnis (`/`) erstellt und die Berechtigungen werden auf 755 gesetzt:

   ```
   # mkdir -m 755 /jumpstart
   ```

4. **Fügen Sie den folgenden Eintrag in die Datei `/etc/dfs/dfstab` ein.**

   `share -F nfs -o ro,anon=0` *Jumpstart-Verz_pfad*

   Mit dem folgenden Eintrag wird zum Beispiel das Verzeichnis `/jumpstart` zur gemeinsamen Nutzung freigegeben:

   ```
   share -F nfs -o ro,anon=0 /jumpstart
   ```

5. **Geben Sie `shareall` ein und drücken Sie die Eingabetaste.**

6. **Entscheiden Sie, ob Sie Beispiele für benutzerdefinierte JumpStart-Dateien in das JumpStart-Verzeichnis kopieren wollen.**

- Wenn nicht, fahren Sie mit Schritt 9 fort.
- Wenn ja, entscheiden Sie anhand der folgenden Tabelle über das weitere Vorgehen.

Beispiel für Speicherorte	Anweisungen
Solaris Operating System-DVD oder Solaris Software-1-CD für die Plattform	Legen Sie die Solaris Operating System-DVD oder die Solaris Software-1-CD in das CD-ROM-Laufwerk des Servers ein. Volume Manager hängt die CD bzw. DVD automatisch ein.
Abbild der Solaris Operating System-DVD oder der Solaris Software-1-CD für die jeweilige Plattform auf einer lokalen Festplatte	Wechseln Sie in das Verzeichnis, in dem sich das Abbild der Solaris Operating System-DVD bzw. der Solaris Software-1-CD befindet. Geben Sie zum Beispiel folgenden Befehl ein: `cd /export/install`

7 Kopieren Sie die Beispieldateien für die benutzerdefinierte JumpStart-Installation in das JumpStart-Verzeichnis auf dem Profilserver.

`# cp -r` *Medienpfad*`/Solaris_10/Misc/jumpstart_sample/*` *Jumpstart-Verz_pfad*

Medienpfad	Der Pfad zu der CD, DVD oder dem Abbild auf der lokalen Festplatte
Jumpstart-Verz_pfad	Der Pfad auf dem Profilserver, auf den Sie die benutzerdefinierten JumpStart-Beispieldateien kopieren

Mit dem folgenden Befehl wird zum Beispiel das Verzeichnis `jumpstart_sample` in das Verzeichnis `/jumpstart` auf dem Profilserver kopiert:

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /jumpstart
```

8 Bearbeiten Sie die JumpStart-Beispieldateien so, dass sie in Ihrer Umgebung funktionieren.

9 Stellen Sie sicher, dass `root` Eigentümer des JumpStart-Verzeichnisses ist und dass die Berechtigungen auf 755 gesetzt sind.

10 Gewähren Sie den Systemen im Netzwerk Zugriff auf den Profilserver.

Weitere Informationen finden Sie unter „So gewähren Sie den Systemen Zugriff auf den Profilserver“ auf Seite 28.

Allen Systemen Zugriff auf den Profilserver gewähren

Wenn Sie einen Profilserver erstellen, müssen Sie sicherstellen, dass die Systeme während der benutzerdefinierten JumpStart-Installation auf das JumpStart-Verzeichnis auf dem Profilserver zugreifen können. Verwenden Sie hierzu eine der folgenden Möglichkeiten:

Befehl oder Datei	Gewähren von Zugriff	Anweisungen
`add_install_client` -Befehl	Verwenden Sie jedes Mal, wenn Sie ein System zur Netzwerkinstallation hinzufügen, den Befehl `add_install_client` mit der Option `-c` , um den Profilserver anzugeben. **Hinweis –** Wenn Sie kein NFS verwenden, müssen Sie ein alternatives Verfahren verwenden, um den Zugriff zu gewähren. ■ **Für SPARC-Systeme** verwenden Sie den Befehl `boot`. ■ **Für x86-Systeme** bearbeiten Sie das GRUB-Menü	■ Bei dem Medium DVD lesen Sie bitte „Hinzufügen der über das Netzwerk zu installierenden Systeme mit einem DVD-Abbild“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation* ■ Bei dem Medium CD lesen Sie bitte „Hinzufügen der über das Netzwerk zu installierenden Systeme mit einem CD-Abbild“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
Geben Sie den Pfad des JumpStart-Verzeichnisses beim Booten des Systems an	■ **Für SPARC-Systeme** verwenden Sie zum Booten des Systems den Befehl `boot`. Geben Sie den Pfad des JumpStart-Verzeichnisses auf dem Profilserver beim Booten des Systems an. Sie müssen die benutzerdefinierten JumpStart-Konfigurationsdateien in eine Datei komprimieren. Speichern Sie dann die komprimierte Konfigurationsdatei auf einem HTTP- oder HTTPS-Server.	■ „Erstellen einer komprimierten Konfigurationsdatei“ auf Seite 69 ■ Schritt 5 in „SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus“ auf Seite 89
	■ **Bei x86-Systemen** geben Sie den Pfad des JumpStart-Verzeichnisses auf dem Profilserver beim Booten des Systems an, indem Sie den Boot-Eintrag im GRUB-Menü bearbeiten. Sie müssen die benutzerdefinierten JumpStart-Konfigurationsdateien in eine Datei komprimieren. Speichern Sie dann die komprimierte Konfigurationsdatei auf einem HTTP- oder HTTPS-Server. Geben Sie beim Bearbeiten des GRUB-Menüeintrags den Speicherort der komprimierten Datei an.	■ „Erstellen einer komprimierten Konfigurationsdatei“ auf Seite 69 ■ „x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB“ auf Seite 96
Datei `/etc/bootparams`	Fügen Sie einen Platzhalter in die Datei `/etc/bootparams` ein.	„So gewähren Sie den Systemen Zugriff auf den Profilserver“ auf Seite 28

▼ So gewähren Sie den Systemen Zugriff auf den Profilserver

Gehen Sie nach dem folgenden Verfahren vor, wenn Sie die Netzwerk-Installationsinformationen an den folgenden Orten speichern möchten:

- In der Datei `/etc/bootparams`.
- In der Namen-Service-Datenbank `bootparams`. Um die Datenbank `bootparams` zu aktualisieren, fügen Sie den in Schritt 3 gezeigten Eintrag ein.

Das folgende Verfahren können Sie nur verwenden, wenn alle Systeme vom gleichen Typ sind (z. B. nur SPARC-Systeme).

Verwenden Sie es nicht unter folgenden Bedingungen:

- Sie speichern das JumpStart-Verzeichnis auf einer Diskette.
- Sie geben die Adresse des Profilservers beim Booten des Systems an. Wenn Sie Systeme mit verschiedenen Architekturen haben, müssen Sie die Adresse des Profilservers beim Booten des Systems angeben.

Wenn die o. g. Bedingungen zutreffen, verwenden Sie den SPARC-Befehl `boot` oder das GRUB-Menü für x86-Systeme.

Hinweis – Sie können die Angaben zur Netzwerkinstallation auch auf einem DHCP-Server ablegen.

- **SPARC-basierte Systeme:** Verwenden Sie den Befehl `add_install_client` mit der Option `-d`, damit das benutzerdefinierte JumpStart-Programm den DHCP-Server verwendet. Alternativ können Sie auch den Befehl `boot` mit der Option `dhcp` verwenden, um das Gleiche zu bewirken. Eine Anleitung zur Verwendung dieser Option finden Sie unter „SPARC: Befehlsreferenz für den Befehl `boot`“ auf Seite 90.
- **Bei x86-Systemen** verwenden Sie dhcp auf eine der folgenden Arten:
  - Wenn Sie einen Installationsserver verwenden, geben Sie den Befehl `add_install_client` mit der Option `-d` ein, um festzulegen, dass das benutzerdefinierte JumpStart-Programm den DHCP-Server mit PXE verwendet.
  - Sie können den GRUB-Eintrag im GRUB-Menü bearbeiten und die dhcp-Option hinzufügen. Eine Anleitung hierzu finden Sie unter „x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB“ auf Seite 96

---

1 Melden Sie sich beim Installations- oder Boot-Server als Superuser an.

2 Öffnen Sie `/etc/bootparams` mit einem Texteditor.

3 Fügen Sie diesen Eintrag hinzu.

`* install_config=`*`Server:Jumpstart-Verz_pfad`*

`*`	Dieser Platzhalter legt fest, dass alle Systeme Zugriff haben.
Server	Der Host-Name des Profilservers, auf dem sich das JumpStart-Verzeichnis befindet.

Jumpstart-Verz_pfad Der absolute Pfad zu dem JumpStart-Verzeichnis.

Der folgende Eintrag gewährt zum Beispiel allen Systemen Zugriff auf das Verzeichnis `/jumpstart` auf dem Profilserver mit dem Namen `sherlock`:

```
* install_config=sherlock:/jumpstart
```

Achtung – Wenn Sie diese Schritte ausgeführt haben, wird beim Booten eines Installationsclients möglicherweise folgende Fehlermeldung ausgegeben:

```
WARNING: getfile: RPC failed: error 5 (RPC Timed out).
```

„Booten vom Netzwerk, Fehlermeldungen“ auf Seite 188 enthält weitere Informationen zu dieser Fehlermeldung.

Jetzt haben alle Systeme Zugriff auf den Profilserver.

Erstellen einer Profildiskette für Standalone-Systeme

Eine Diskette, auf der sich ein JumpStart-Verzeichnis befindet, wird als Profildiskette bezeichnet. Ein System, das nicht mit dem Netzwerk verbunden ist, hat keinen Zugriff auf einen Profilserver. Sie müssen deshalb das JumpStart-Verzeichnis auf eine Diskette kopieren, wenn ein System nicht an das Netzwerk angeschlossen ist. Das System, auf dem Sie die Profildiskette erstellen, muss über ein Diskettenlaufwerk verfügen.

Das JumpStart-Verzeichnis enthält alle wichtigen benutzerdefinierten JumpStart-Dateien, wie zum Beispiel die Datei `rules`, die Datei `rules.ok` und Profile. Das JumpStart-Verzeichnis muss im Root-Verzeichnis (`/`) der Profildiskette angelegt werden.

Siehe eines der folgenden Verfahren:

- „SPARC: So erstellen Sie eine Profildiskette“ auf Seite 30
- „x86: So erstellen Sie eine Profildiskette mit GRUB“ auf Seite 32

▼ SPARC: So erstellen Sie eine Profildiskette

Hinweis – Bei diesem Verfahren wird davon ausgegangen, dass Volume Manager auf dem System ausgeführt wird. Wenn Sie zum Verwalten von Disketten, CDs und DVDs nicht Volume Manager verwenden, finden Sie Näheres zum Verwalten von Wechseldatenträgern ohne Volume Manager im *System Administration Guide: Devices and File Systems*.

1. **Suchen Sie ein SPARC-System mit angeschlossenem Diskettenlaufwerk.**

2. **Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.**

   Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

3. **Legen Sie eine leere Diskette oder eine Diskette, die überschrieben werden kann, in das Diskettenlaufwerk ein.**

4. **Hängen Sie die Diskette ein.**

   ```
   # volcheck
   ```

5. **Stellen Sie fest, ob die Diskette ein UNIX-Dateisystem (UFS) enthält.**

   Überprüfen Sie, ob die Datei `/etc/mnttab` auf dem System einen Eintrag wie den Folgenden enthält:

   ```
   /vol/dev/diskette0/scrap  /floppy/scrap  ufs  suid,rw,largefiles,dev=1740008  927147040
   ```

   - Wenn der Eintrag vorhanden ist, fahren Sie mit Schritt 7 fort.
   - Wenn der Eintrag nicht vorhanden ist, fahren Sie mit dem nächsten Schritt fort.

6. **Erstellen Sie ein UFS auf der Diskette.**

   ```
   # newfs /vol/dev/aliases/floppy0
   ```

7. **Entscheiden Sie, ob Sie Beispiele für benutzerdefinierte JumpStart-Dateien in das JumpStart-Verzeichnis kopieren wollen.**

   - Wenn nicht, fahren Sie mit Schritt 10 fort.
   - Wenn ja, entscheiden Sie anhand der folgenden Tabelle über das weitere Vorgehen.

Beispiel für Speicherorte	Anweisungen
Solaris Operating System for SPARC Platforms-DVD oder Solaris Software for SPARC Platforms - 1-CD	Legen Sie die Solaris Operating System for SPARC Platforms-DVD oder die Solaris Software for SPARC Platforms - 1-CD in das CD-ROM-Laufwerk des Servers ein. Volume Manager hängt die CD bzw. DVD automatisch ein.
Ein Abbild der Solaris Operating System for SPARC Platforms-DVD oder der Solaris Software for SPARC Platforms - 1-CD auf einer lokalen Festplatte	Wechseln Sie in das Verzeichnis, in dem sich das Abbild der Solaris Operating System for SPARC Platforms-DVD oder der Solaris Software for SPARC Platforms - 1-CD befindet. Geben Sie zum Beispiel folgenden Befehl ein: `cd /export/install`

8 **Kopieren Sie die angepassten JumpStart-Beispieldateien in das JumpStart-Verzeichnis auf der Profildiskette.**

```
# cp -r Medienpfad/Solaris_10/Misc/jumpstart_sample/* Jumpstart-Verz_pfad
```

Medienpfad	Der Pfad zu der CD, DVD oder dem Abbild auf der lokalen Festplatte
Jumpstart-Verz_pfad	Der Pfad zu der Profildiskette, auf die Sie die benutzerdefinierten JumpStart-Beispieldateien kopieren möchten

Hinweis – Sie müssen alle benutzerdefinierten JumpStart-Installationsdateien in das Root-Verzeichnis (`/`) auf der Diskette stellen.

Mit dem folgenden Befehl wird zum Beispiel der Inhalt von `jumpstart_sample` auf der Solaris Software for SPARC Platforms - 1-CD in das Root-Verzeichnis (`/`) auf einer Profildiskette mit dem Namen `scrap` kopiert:

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

9 **Bearbeiten Sie die JumpStart-Beispieldateien auf der Profildiskette so, dass sie in Ihrer Umgebung funktionieren.**

10 **Stellen Sie sicher, dass `root` Eigentümer des JumpStart-Verzeichnisses ist und dass die Berechtigungen auf 755 gesetzt sind.**

11 **Lassen Sie die Diskette auswerfen.**

```
# eject floppy
```

Damit ist die Erstellung einer Profildiskette abgeschlossen. Jetzt können Sie die Datei `rules` bearbeiten und Profile auf der Profildiskette erstellen, um benutzerdefinierte JumpStart-Installationen zu definieren. Das weitere Vorgehen ist unter „Erstellen der Datei `rules`“ auf Seite 34 beschrieben.

▼ x86: So erstellen Sie eine Profildiskette mit GRUB

Führen Sie die folgenden Schritte aus, um eine Profildiskette mit GRUB zu erstellen. Bei der Installation wird ein GRUB-Menü angeboten, das den Bootvorgang übernimmt. Das GRUB-Menü ersetzt die Solaris Device Configuration Assistant, die in älteren Releases u. U. zum Booten eines Systems benötigt wurde.

Hinweis – Bei diesem Verfahren wird davon ausgegangen, dass Volume Manager auf dem System ausgeführt wird. Wenn Sie zum Verwalten von Disketten, CDs und DVDs nicht Volume Manager verwenden, finden Sie Näheres zum Verwalten von Wechseldatenträgern ohne Volume Manager im *System Administration Guide: Devices and File Systems*.

1 **Suchen Sie ein x86-System mit angeschlossenem Diskettenlaufwerk.**

2 **Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.**

Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

3 **Legen Sie eine leere Diskette oder eine Diskette, die überschrieben werden kann, in das Diskettenlaufwerk ein.**

4 **Hängen Sie die Diskette ein.**

```
# volcheck
```

5 **Entscheiden Sie, ob Sie Beispiele für benutzerdefinierte JumpStart-Dateien in das JumpStart-Verzeichnis kopieren wollen.**

- Wenn nicht, fahren Sie mit Schritt 8 fort.
- Wenn ja, entscheiden Sie anhand der folgenden Tabelle über das weitere Vorgehen.

Beispiel für Speicherorte	Anweisungen
Solaris Operating System for x86 Platforms-DVD oder Solaris Software for x86 Platforms - 1-CD	Legen Sie die Solaris Operating System for x86 Platforms-DVD oder die Solaris Software for x86 Platforms - 1-CD in das CD-ROM-Laufwerk des Servers ein. Volume Manager hängt die DVD·bzw. CD automatisch ein.
Ein Abbild der Solaris Operating System for x86 Platforms-DVD oder der Solaris Software for x86 Platforms - 1-CD auf einer lokalen Festplatte	Wechseln Sie in das Verzeichnis, in dem sich das Abbild der Solaris Operating System for x86 Platforms-DVD oder der Solaris Software for x86 Platforms - 1-CD befindet. Geben Sie zum Beispiel Folgendes ein: `cd /export/install`

6 **Kopieren Sie die angepassten JumpStart-Beispieldateien in das JumpStart-Verzeichnis auf der Profildiskette.**

```
# cp -r Medienpfad/Solaris_10/Misc/jumpstart_sample/* Jumpstart-Verz_pfad
```

Medienpfad Der Pfad zu der CD, DVD oder dem Abbild auf der lokalen Festplatte

Jumpstart-Verz_pfad Der Pfad zu der Profildiskette, auf die Sie die benutzerdefinierten JumpStart-Beispieldateien kopieren möchten

---

Hinweis – Sie müssen alle benutzerdefinierten JumpStart-Installationsdateien in das Root-Verzeichnis (/) auf der Profildiskette stellen.

Mit dem folgenden Befehl wird zum Beispiel der Inhalt von `jumpstart_sample` auf der Solaris Software for x86 Platforms - 1-CD in das Root-Verzeichnis (/) auf einer Profildiskette mit dem Namen `scrap` kopiert:

```
cp -r /cdrom/cdrom0/Solaris_10/Misc/jumpstart_sample/* /floppy/scrap
```

7 Bearbeiten Sie die JumpStart-Beispieldateien auf der Profildiskette so, dass sie in Ihrer Umgebung funktionieren.

8 Stellen Sie sicher, dass `root` Eigentümer des JumpStart-Verzeichnisses ist und dass die Berechtigungen auf 755 gesetzt sind.

9 Lassen Sie die Diskette auswerfen, indem Sie im Fenster „Dateimanager" auf „Datenträger auswerfen" klicken, oder geben Sie in der Befehlszeile den Befehl `eject floppy` ein.

10 Klicken Sie im Dialogfeld „Manager für Wechsel-Datenträger" auf OK.

11 Lassen Sie die Diskette von Hand auswerfen.

Siehe auch Damit ist die Erstellung einer Profildiskette abgeschlossen. Jetzt können Sie die Datei `rules` bearbeiten und Profile auf der Profildiskette erstellen, um benutzerdefinierte JumpStart-Installationen zu definieren. Das weitere Vorgehen ist unter „Erstellen der Datei `rules`" auf Seite 34 beschrieben.

Erstellen der Datei `rules`

Die Datei `rules` ist eine Textdatei, die eine Regel (rule) für jede Systemgruppe enthält, auf der Sie die Betriebssystemumgebung Solaris installieren wollen. Jede Regel charakterisiert eine Gruppe von Systemen auf der Grundlage von einem oder mehreren Systemattributen. Jede Regel verknüpft außerdem jede Gruppe mit einem Profil. Ein Profil ist eine Textdatei, in der definiert ist, wie die Solaris-Software auf den Systemen in der Gruppe installiert werden soll. Die folgende Regel legt zum Beispiel fest, dass das JumpStart-Programm die Informationen im Profil `basic_prof` zur Installation aller Systeme der Plattformgruppe `sun4u` verwenden soll.

```
karch sun4u - basic_prof -
```

Die Datei rules dient zum Generieren der Datei rules.ok, die für benutzerdefinierte JumpStart-Installationen erforderlich ist.

Hinweis – Wenn Sie das JumpStart-Verzeichnis wie unter „Erstellen einer Profildiskette für Standalone-Systeme“ auf Seite 30 oder „Erstellen eines Profilservers für vernetzte Systeme“ auf Seite 25 erläutert einrichten, befindet sich bereits eine rules-Beispieldatei im JumpStart-Verzeichnis. Diese rules-Beispieldatei enthält Dokumentation und einige Beispielregeln. Wenn Sie mit der rules-Beispieldatei arbeiten, müssen Sie die nicht verwendeten Beispielregeln auf Kommentar setzen.

Syntax der Datei rules

Die Datei rules muss folgende Merkmale aufweisen:

- Die Datei muss den Namen rules erhalten.
- Die Datei muss mindestens eine Regel enthalten.

Die rules-Datei kann Folgendes enthalten:

- Auf Kommentar gesetzten Text

  Text, der in einer Zeile hinter dem Zeichen # steht, wird von JumpStart als Kommentar interpretiert. Wenn eine Zeile mit dem Symbol # anfängt, wird die gesamte Zeile als Kommentar behandelt.
- Eine oder mehrere Leerzeilen
- Eine oder mehrere Regeln

  Um eine Regel in der Folgezeile fortzuführen, geben Sie vor dem Zeilenwechsel einen umgekehrten Schrägstrich (Backslash: \) ein, bevor Sie die Eingabetaste drücken.

▼ So erstellen Sie die Datei rules

1 **Erstellen Sie in einem Texteditor eine Datei mit dem Namen rules. Anderenfalls öffnen Sie in dem von Ihnen erstellten JumpStart-Verzeichnis eine rules-Beispieldatei.**

2 **Fügen Sie für jede Gruppe von Systemen, auf denen Sie die Solaris-Software installieren wollen, eine Regel in die Datei rules ein.**

Eine Liste der Schlüsselwörter und Werte für die Datei rules finden Sie unter „Rule-Schlüsselwörter und -Werte“ auf Seite 111.

Eine Regel in einer rules-Datei muss die folgende Syntax einhalten:

!*Regelschlüsselwort Regelwert* && !*Regelschlüsselwort Regelwert* ... *Beginn Profil Ende*

!	Ein Symbol, das vor einem Schlüsselwort verwendet wird, um eine Negation anzugeben.
Schlüsselwort	Eine vordefinierte lexikalische Einheit oder ein Wort, die bzw. das ein allgemeines Systemattribut beschreibt, wie zum Beispiel den Host-Namen (`hostname`) oder die Hauptspeichergröße (`memsize`). Das *Schlüsselwort* wird zusammen mit dem Rule-Wert verwendet, um die Attribute zu definieren, die ein System aufweisen muss, damit das Profil darauf angewendet wird. Eine Liste der Rule-Schlüsselwörter finden Sie unter „Rule-Schlüsselwörter und -Werte“ auf Seite 111.
Regelwert	Ein Wert, der das jeweilige Systemattribut für das zugehörige Rule-Schlüsselwort angibt. Rule-Werte sind unter „Rule-Schlüsselwörter und -Werte“ auf Seite 111.
&&	Ein Symbol, mit dem Sie Paare von Rule-Schlüsselwörtern und Rule-Werten in einer Regel zusammenfassen können (logisches UND). Während einer benutzerdefinierten JumpStart-Installation werden nur Systeme, die allen Paaren in der Regel entsprechen, als der Regel entsprechend erkannt.
Beginn	Der Name eines optionalen Bourne-Shell-Skripts, das vor der Installation ausgeführt werden kann. Wenn kein Begin-Skript vorhanden ist, müssen Sie in dieses Feld ein Minuszeichen (-) eingeben. Alle Begin-Skripte müssen sich im JumpStart-Verzeichnis befinden. Informationen zum Erstellen von Begin-Skripten finden Sie unter „Erstellen von Begin-Skripten“ auf Seite 59.
Profil	Der Name einer Textdatei, in der definiert ist, wie die Solaris-Software auf dem System installiert werden soll, wenn ein System der Regel entspricht. Die Informationen in einem Profil bestehen aus Profilschlüsselwörtern und den entsprechenden Profilwerten. Alle Profile müssen sich im JumpStart-Verzeichnis befinden. **Hinweis –** Weitere Möglichkeiten, das Profilfeld zu verwenden, sind unter „Verwenden eines standortspezifischen Installationsprogramms“ auf Seite 76 und „Erstellen abgeleiteter Profile mit einem Begin-Skript“ auf Seite 60.
Ende	Der Name eines optionalen Bourne-Shell-Skripts, das nach der Installation ausgeführt werden kann. Wenn kein Finish-Skript vorhanden ist, müssen Sie in dieses Feld ein Minuszeichen (-) eingeben. Alle Finish-Skripte müssen sich im JumpStart-Verzeichnis befinden.

Informationen zum Erstellen von Finish-Skripten finden Sie unter „Erstellen von Finish-Skripten“ auf Seite 62.

Jede Regel muss mindestens Folgendes enthalten:

- Ein Schlüsselwort, einen Wert und ein zugehöriges Profil
- Ein Minuszeichen (-) im *Beginn*- und *Ende*-Feld, wenn kein Begin- bzw. Finish-Skript angegeben wird

3 **Speichern Sie die Datei `rules` im JumpStart-Verzeichnis.**

4 **Stellen Sie sicher, dass `root` Eigentümer der Datei `rules` ist und dass die Berechtigungen auf 644 gesetzt sind.**

`rules`-Beispieldatei

Das folgende Beispiel zeigt mehrere Beispiele für Regeln in einer `rules`-Datei. Jede Zeile enthält ein Rule-Schlüsselwort und einen gültigen Wert für dieses Schlüsselwort. Das JumpStart-Programm durchsucht die Datei `rules` von oben nach unten.

Wenn das JumpStart-Programm ein Rule-Schlüsselwort und einen Rule-Wert mit einem bekannten System vergleicht und eine Übereinstimmung erkennt, installiert das JumpStart-Programm die Solaris-Software so, wie dies in dem im Profilfeld angegeben Profil festgelegt ist.

Eine vollständige Liste mit Einschränkungen der Datei `rules` finden Sie in „Syntax der Datei `rules`“ auf Seite 35.

BEISPIEL 3–1 `rules`-Datei

```
# rule keywords and rule values       begin script       profile       finish script
# ------------------------------       ------------       --------      -------------
  hostname eng-1                      -                  basic_prof    -
  network 192.168.255.255 && !model \
 'SUNW,Sun-Blade-100'                 -                  net_prof      -
  model SUNW,SPARCstation-LX          -                  lx_prof       complete
  network 192.168.2.0 && karch i86pc  setup              x86_prof      done
  memsize 64-128 && arch i386         -                  prog_prof     -
  any   -                             -                  generic_prof  -
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`Host-Name` Ein System entspricht der Regel, wenn der Host-Name `eng-1` lautet. Zur Installation der Solaris-Software auf einem System, das dieser Regel entspricht, wird das Profil `basic_prof` verwendet.

BEISPIEL 3–1 `rules`-Datei *(Fortsetzung)*

`network`	Ein System entspricht der Regel, wenn es sich in Teilnetz 192.168.255.255 befindet und es sich *nicht* um eine Sun Blade ™ 100 (`SUNW,Sun-Blade-100`) handelt. Zur Installation der Solaris-Software auf Systemen, die dieser Regel entsprechen, wird das Profil `net_prof` verwendet. Diese Regel ist auch ein Beispiel für die Weiterführung einer einzelnen Regel in einer neuen Zeile mithilfe eines umgekehrten Schrägstrichs (\).
`model`	Ein System entspricht der Regel, wenn es sich um eine SPARCstation LX handelt. Zur Installation der Solaris-Software auf Systemen, die dieser Regel entsprechen, werden das Profil `lx_prof` und das Finish-Skript `complete` verwendet.
`network`	Ein System entspricht der Regel, wenn es sich in Teilnetz 192.168.2.0 befindet und es sich um ein x86-basiertes sun4u-System handelt. Zur Installation der Solaris-Software auf Systemen, die dieser Regel entsprechen, werden das Begin-Skript `setup`, das Profil `x864u_prof` und das Finish-Skript `done` verwendet.
`memsize`	Ein System entspricht der Regel, wenn es zwischen 64 und 128 MB Hauptspeicher hatund es sich um ein x86-basiertes System handelt. Zur Installation der Solaris-Software auf Systemen, die dieser Regel entsprechen, wird das Profil `prog_prof` verwendet.
`any`	Diese Regel gilt für alle Systeme, die keiner der oben genannten Regeln entsprechen. Zur Installation der Solaris-Software auf Systemen, die dieser Regel entsprechen, wird das Profil `generic_prof` verwendet. Wenn `any` verwendet wird, sollte dies immer die letzte Regel in der Datei `rules` sein.

Erstellen eines Profils

Ein Profil ist eine Textdatei, in der definiert ist, wie die Solaris-Software auf einem System installiert wird. Ein Profil definiert Elemente der Installation, wie zum Beispiel die zu installierende Softwaregruppe. Jede Regel gibt ein Profil an, das definiert, wie ein System installiert werden soll. Sie können ein eigenes Profil für jede Regel erstellen oder ein Profil in mehreren Regeln verwenden.

Ein Profil besteht aus einem oder mehreren Profilschlüsselwörtern und den dazugehörigen Werten. Jedes Profilschlüsselwort ist ein Befehl, der einen Aspekt der Installation der Solaris-Software durch das JumpStart-Programm regelt. Mit dem folgenden Profilschlüsselwort und -wert wird zum Beispiel festgelegt, dass das System vom JumpStart-Programm als ein Server installiert werden soll:

```
system_type  server
```

Hinweis – Wenn Sie das JumpStart-Verzeichnis wie in einem der folgenden Verfahren erläutert erstellt haben, befinden sich darin bereits Beispielprofile:

- „Erstellen eines Profilservers für vernetzte Systeme“ auf Seite 25
- „Erstellen einer Profildiskette für Standalone-Systeme“ auf Seite 30

Syntax von Profilen

Ein Profil muss Folgendes enthalten:

- Das Schlüsselwort `install_type` als ersten Eintrag
- Ein Schlüsselwort pro Zeile
- Das Schlüsselwort `root_device`, wenn die Systeme, auf denen mithilfe des Profils ein Upgrade ausgeführt werden soll, mehr als ein Root-Dateisystem (`/`) enthalten, in dem das Upgrade ausgeführt werden kann

Ein Profil kann Folgendes enthalten:

- Auf Kommentar gesetzten Text

  Text, der in einer Zeile nach dem Symbol # erscheint, wird vom JumpStart-Programm als Kommentar behandelt. Wenn eine Zeile mit dem Symbol # anfängt, wird die gesamte Zeile als Kommentar behandelt.
- Eine oder mehrere Leerzeilen

▼ So erstellen Sie ein Profil

1 **Erstellen Sie in einem Texteditor eine Textdatei. Geben Sie der Datei einen aussagekräftigen Namen. Anderenfalls öffnen Sie in dem von Ihnen erstellten JumpStart-Verzeichnis eine Beispieldatei.**

Hinweis – Stellen Sie sicher, dass der Name des Profils wiedergibt, wie Sie das Profil zum Installieren der Solaris-Software auf einem System einsetzen wollen. So können Sie zum Beispiel die Profile `basic_install`, `eng_profile` oder `user_profile` anlegen.

2 **Fügen Sie Schlüsselwörter und Werte zu dem Profil hinzu.**

Eine Liste der Schlüsselwörter und Werte finden Sie unter „Profilschlüsselwörter und -werte“ auf Seite 116.

Hinweis – Bei Profilschlüsselwörtern und deren Werten wird zwischen Groß- und Kleinschreibung unterschieden.

3 **Speichern Sie das Profil im JumpStart-Verzeichnis.**

4 **Stellen Sie sicher, dass `root` Eigentümer des Profils ist und dass die Berechtigungen auf 644 gesetzt sind.**

5 **Testen Sie das Profil (optional).**

„Testen eines Profils“ auf Seite 52 enthält Informationen zum Testen von Profilen.

Beispiele für Profile

Die folgenden Beispiele für Profile zeigen, wie Sie mit verschiedenen Profilschlüsselwörtern und Werten steuern können, wie die Solaris-Software auf einem System installiert wird. „Profilschlüsselwörter und -werte“ auf Seite 116 enthält eine Beschreibung der Profilschlüsselwörter und Werte.

Hinweis – Wenn Sie ein Solaris-ZFS™-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

BEISPIEL 3–2 Einhängen von entfernten Dateisystemen und Hinzufügen und Entfernen von Packages

```
# profile keywords        profile values
# -----------------       -----------------
  install_type            initial_install
  system_type             standalone
  partitioning            default
  filesys                 any 512 swap   # specify size of /swap
  cluster                 SUNWCprog
  package                 SUNWman delete
  cluster                 SUNWCacc
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type` Das Schlüsselwort `install_type` ist in jedem Profil erforderlich.

`system_type` Das Schlüsselwort `system_type` legt fest, dass die Installation auf einem Standalone-System erfolgen soll.

BEISPIEL 3–2 Einhängen von entfernten Dateisystemen und Hinzufügen und Entfernen von Packages *(Fortsetzung)*

`partitioning` Der Wert `default` legt fest, dass die Dateisystem-Slices von der zu installierenden Software definiert werden. Die Größe des `swap`-Bereichs wird auf 512 MB festgelegt und dieser Bereich wird auf einer beliebigen Festplatte (Wert `any`) angelegt.

`cluster` Die Solaris-Softwaregruppe für Entwickler, `SUNWCprog`, wird auf dem System installiert.

`package` Werden die Standard-Manpages vom Dateiserver im Netzwerk, `s_ref`, eingehängt, dann sind die Manpage-Packages nicht auf dem System zu installieren. Die Packages, die die System Accounting-Dienstprogramme enthalten, sind für die Installation auf dem System ausgewählt.

BEISPIEL 3–3 Einhängen von entfernten Dateisystemen und Hinzufügen von Packages von Drittanbietern

```
# profile keywords        profile values
# -----------------       -----------------
  install_type            initial_install
  system_type             standalone
  partitioning            default
  filesys                 any 512 swap   # specify size of /swap
  cluster                 SUNWCprog
  cluster                 SUNWCacc
  package                 apache_server  \
                           http://package.central/packages/apache timeout 5
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type` Das Schlüsselwort `install_type` ist in jedem Profil erforderlich.

`system_type` Das Schlüsselwort `system_type` legt fest, dass die Installation auf einem Standalone-System erfolgen soll.

`partitioning` Der Wert `default` legt fest, dass die Dateisystem-Slices von der zu installierenden Software definiert werden. Die Größe des `swap`-Bereichs wird auf 512 MB festgelegt und dieser Bereich wird auf einer beliebigen Festplatte (Wert `any`) angelegt.

`cluster` Die Solaris-Softwaregruppe für Entwickler, `SUNWCprog`, wird auf dem System installiert.

`package` Ein fremdes Package von einem HTTP-Server wird auf dem System installiert.

BEISPIEL 3–4 Festlegen des Installationsorts von Dateisystemen

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  system_type             standalone
  partitioning            explicit
  filesys                 c0t0d0s0 auto /
  filesys                 c0t3d0s1 auto swap
  filesys                 any auto usr
  cluster                 SUNWCall
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`partitioning` Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden. Die Größe des Root-Dateisystems (`/`) basiert auf der ausgewählten Software (Wert `auto`), und das Root-Dateisystem wird auf `c0t0d0s0` angelegt. Der `swap`-Bereich wird auf `c0t3d0s1` angelegt und seine Größe nach Bedarf automatisch festgelegt. `usr` basiert auf der ausgewählten Software und das Installationsprogramm ermittelt auf der Grundlage des Werts `any`, wo `usr` angelegt wird.

`cluster` Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert.

BEISPIEL 3–5 Upgrade/Installieren von Patches

```
# profile keywords         profile values
# ----------------         -------------------
  install_type             upgrade
  root_device              c0t3d0s2
  backup_media             remote_filesystem timber:/export/scratch
  package                  SUNWbcp delete
  package                  SUNWxwman add
  cluster                  SUNWCacc add
  patch                    patch_list nfs://patch_master/Solaris_10/patches \
                           retry 5
  locale                   de
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type` Dieses Profil bewirkt, dass beim Upgrade des Systems eine Neuzuweisung des Festplattenspeichers erfolgt. In diesem Beispiel muss Festplattenspeicher neu zugewiesen werden, da in einigen Dateisystemen nicht genug Platz für das Upgrade vorhanden ist.

BEISPIEL 3–5 Upgrade/Installieren von Patches *(Fortsetzung)*

`root_device`	Das Upgrade erfolgt im Root-Dateisystem auf `c0t3d0s2`.
`backup_media`	Ein entferntes System mit dem Namen `timber` wird beim Neuzuweisen von Festplattenspeicher zum Sichern der Daten verwendet. Weitere Werte für Speicherdatenträger-Schlüsselwörter finden Sie unter „`backup_media`-Profilschlüsselwort“ auf Seite 124.
`package`	Das Package für die Binärkompatibilität, `SUNWbcp`, wird nach dem Upgrade nicht auf dem System installiert.
`package`	Hiermit wird bewirkt, dass die Manpages zu X Window System sowie die System Accounting-Dienstprogramme auf dem System installiert werden, sofern sie noch nicht installiert sind. Alle bereits auf dem System installierten Packages·werden automatisch aktualisiert.
`patch`	Eine Liste der Patches, die beim Upgrade installiert werden. Die Patchliste befindet sich auf dem NFS-Server `patch_master` im Verzeichnis `Solaris_10/patches`. Sollte das NFS-Einhängen fehlschlagen, werden maximal fünf Versuche unternommen.
`Sprachumgebung`	Die deutschen Lokalisierungs-Packages werden auf dem System installiert.

BEISPIEL 3–6 Neuzuweisen von Festplattenspeicher für ein Upgrade

```
# profile keywords         profile values
# ----------------         -------------------
  install_type             upgrade
  root_device              c0t3d0s2
  backup_media             remote_filesystem timber:/export/scratch
  layout_constraint        c0t3d0s2 changeable 100
  layout_constraint        c0t3d0s4 changeable
  layout_constraint        c0t3d0s5 movable
  package                  SUNWbcp delete
  package                  SUNWxwman add
  cluster                  SUNWCacc add
  locale                   de
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type`	Dieses Profil bewirkt, dass beim Upgrade des Systems eine Neuzuweisung des Festplattenspeichers erfolgt. In diesem Beispiel muss Festplattenspeicher neu zugewiesen werden, da in einigen Dateisystemen nicht genug Platz für das Upgrade vorhanden ist.
`root_device`	Das Upgrade erfolgt im Root-Dateisystem auf `c0t3d0s2`.

BEISPIEL 3–6 Neuzuweisen von Festplattenspeicher für ein Upgrade *(Fortsetzung)*

`backup_media`	Ein entferntes System mit dem Namen `timber` wird beim Neuzuweisen von Festplattenspeicher zum Sichern der Daten verwendet. Weitere Werte für Speicherdatenträger-Schlüsselwörter finden Sie unter „`backup_media`-Profilschlüsselwort“ auf Seite 124.
`layout_constraint`	Die `layout_constraint`-Schlüsselwörter legen fest, dass Auto-Layout beim Neuzuweisen von Festplattenspeicher für das Upgrade folgende Funktionen ausführen kann. ■ Ändern der Slices 2 und 4. Die Slices können an eine andere Adresse verschoben und ihre Größe kann geändert werden. ■ Verschieben von Slice 5. Das Slice kann verschoben werden, aber die Größe darf sich nicht ändern.
`package`	Das Package für die Binärkompatibilität, `SUNWbcp`, wird nach dem Upgrade nicht auf dem System installiert.
`package`	Hiermit wird bewirkt, dass die Manpages zu X Window System sowie die System Accounting-Dienstprogramme auf dem System installiert werden, sofern sie noch nicht installiert sind. Alle bereits auf dem System installierten Packages·werden automatisch aktualisiert.
`Sprachumgebung`	Die deutschen Lokalisierungs-Packages werden auf dem System installiert.

BEISPIEL 3–7 Abrufen eines Solaris Flash-Archivs von einem HTTP-Server

Im folgenden Beispiel gibt das Profil an, dass das benutzerdefinierte JumpStart-Programm das Solaris Flash-Archiv von einem HTTP-Server abruft.

```
# profile keywords         profile values
# ----------------         -------------------
install_type               flash_install
archive_location           http://192.168.255.255/flasharchive/solarisarchive
partitioning               explicit
filesys                    c0t1d0s0 4000 /
filesys                    c0t1d0s1 512 swap
filesys                    c0t1d0s7 free /export/home
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type`	Das Profil installiert ein Solaris Flash-Archiv auf dem Klonsystem. Wie bei einer Erst- bzw. Neuinstallation werden alle Dateien überschrieben.
`archive_location`	Das Solaris Flash-Archiv wird von einem HTTP-Server abgerufen.

BEISPIEL 3–7 Abrufen eines Solaris Flash-Archivs von einem HTTP-Server *(Fortsetzung)*

`partitioning` Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden. Die Größe von Root (`/`) ist von der Größe des Solaris Flash-Archivs abhängig. Das Root-Dateisystem wird auf `c0t1d0s0` angelegt. Der `swap`-Bereich wird auf `c0t1d0s1` angelegt und seine Größe nach Bedarf automatisch festgelegt. `/export/home` ist vom verbleibenden Speicherplatz abhängig. `/export/home` wird auf `c0t1d0s7` angelegt.

BEISPIEL 3–8 Abrufen eines Solaris Flash-Archivs von einem sicheren HTTP-Server

Das Profil im folgenden Beispiel sieht vor, dass das Programm für die benutzerdefinierte JumpStart-Installation das Solaris Flash-Archiv von einem sicheren HTTP-Server abruft.

```
# profile keywords        profile values
# ----------------        -------------------
install_type              flash_install
archive_location          https://192.168.255.255/solarisupdate.flar
partitioning              explicit
filesys                   c0t1d0s0 4000 /
filesys                   c0t1d0s1 512 swap
filesys                   c0t1d0s7 free /export/home
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type` Das Profil installiert ein Solaris Flash-Archiv auf dem Klonsystem. Wie bei einer Erst- bzw. Neuinstallation werden alle Dateien überschrieben.

`archive_location` Das komprimierte Solaris Flash-Archiv wird von einem sicheren HTTP-Server abgerufen.

`partitioning` Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden. Die Größe von Root (`/`) ist von der Größe des Solaris Flash-Archivs abhängig. Der `swap`-Bereich wird auf `c0t1d0s1` angelegt und seine Größe nach Bedarf automatisch festgelegt. `/export/home` ist vom verbleibenden Speicherplatz abhängig. `/export/home` wird auf `c0t1d0s7` angelegt.

BEISPIEL 3–9 Abrufen eines Solaris Flash-Archivs und Installieren eines Packages von einem Drittanbieter

Im folgenden Beispiel gibt das Profil an, dass das benutzerdefinierte JumpStart-Programm das Solaris Flash-Archiv von einem HTTP-Server abruft.

BEISPIEL 3–9 Abrufen eines Solaris Flash-Archivs und Installieren eines Packages von einem Drittanbieter *(Fortsetzung)*

```
# profile keywords        profile values
# ----------------        -------------------
install_type              flash_install
archive_location          http://192.168.255.255/flasharchive/solarisarchive
partitioning              explicit
filesys                   c0t1d0s0 4000 /
filesys                   c0t1d0s1 512 swap
filesys                   c0t1d0s7 free /export/home
package                   SUNWnew http://192.168.254.255/Solaris_10 timeout 5
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type` Das Profil installiert ein Solaris Flash-Archiv auf dem Klonsystem. Wie bei einer Erst- bzw. Neuinstallation werden alle Dateien überschrieben.

`archive_location` Das Solaris Flash-Archiv wird von einem HTTP-Server abgerufen.

`partitioning` Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden. Die Größe von Root (`/`) ist von der Größe des Solaris Flash-Archivs abhängig. Das Root-Dateisystem wird auf `c0t1d0s0` angelegt. Der `swap`-Bereich wird auf `c0t1d0s1` angelegt und seine Größe nach Bedarf automatisch festgelegt. `/export/home` ist vom verbleibenden Speicherplatz abhängig. `/export/home` wird auf `c0t1d0s7` angelegt.

`package` Aus dem Verzeichnis `Solaris_10` auf dem HTTP-Server `192.168.254.255` wird das Package SUNWnew hinzugefügt.

BEISPIEL 3–10 Abrufen eines Solaris Flash-Differenzarchivs von einem NFS-Server

Im folgenden Beispiel gibt das Profil an, dass das benutzerdefinierte JumpStart-Programm das Solaris Flash-Archiv von einem NFS-Server abruft. Das Schlüsselwort `flash_update` gibt an, dass es sich hierbei um ein Differenzarchiv handelt. Ein Differenzarchiv installiert nur die Unterschiede zwischen zwei Systemabbildern.

```
# profile keywords        profile values
# ----------------        -------------------
install_type              flash_update
archive_location          nfs installserver:/export/solaris/flasharchive \
                          /solarisdiffarchive
no_master_check
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

BEISPIEL 3–10 Abrufen eines Solaris Flash-Differenzarchivs von einem NFS-Server *(Fortsetzung)*

`install_type`	Das Profil installiert ein Solaris Flash-Differenzarchiv auf dem Klonsystem. Es werden nur die im Archiv aufgeführten Dateien installiert.
`archive_location`	Das Solaris Flash-Archiv wird von einem NFS-Server abgerufen.
`no_master_check`	Das Klonsystem wird nicht auf das Vorhandensein eines gültigen Systemabbilds überprüft. Ein gültiges Systemabbild ist ein vom Original-Mastersystem hergestelltes.

BEISPIEL 3–11 Erstellen einer leeren Boot-Umgebung

Im folgenden Beispiel gibt das Profil an, dass das benutzerdefinierte JumpStart-Programm eine leere Boot-Umgebung erstellen soll. Eine leere Boot-Umgebung enthält keine Dateisysteme, und es werden keine Daten aus der aktuellen Boot-Umgebung kopiert. Die Boot-Umgebung kann später mit einem Solaris Flash-Archiv bespielt und aktiviert werden.

```
# profile keywords        profile values
# ----------------        ------------------
  install_type            initial_install
  system_type             standalone
  partitioning            explicit
  filesys                 c0t0d0s0 auto /
  filesys                 c0t3d0s1 auto swap
  filesys                 any auto usr
  cluster                 SUNWCall
  bootenv createbe bename second_BE \
  filesystem /:/dev/dsk/c0t1d0s0:ufs \
  filesystem -:/dev/dsk/c0t1d0s0:swap \
  filesystem /export:shared:ufs
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`partitioning`	Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden. Die Größe des Root-Dateisystems (`/`) basiert auf der ausgewählten Software (Wert `auto`), und das Root-Dateisystem wird auf `c0t0d0s0` angelegt. Der `swap`-Bereich wird auf `c0t3d0s1` angelegt und seine Größe nach Bedarf automatisch festgelegt. `usr` basiert auf der ausgewählten Software und das Installationsprogramm ermittelt auf der Grundlage des Werts `any`, wo `usr` angelegt wird.
`cluster`	Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert.

BEISPIEL 3–11 Erstellen einer leeren Boot-Umgebung *(Fortsetzung)*

`bootenv createbe` Auf der Festplatte `c0t1d0` wird eine leere, inaktive Boot-Umgebung angelegt. Die Dateisysteme für Root (`/`), Swap und `/export` werden angelegt, bleiben jedoch leer. Diese zweite Boot-Umgebung kann später mit einem Solaris Flash-Archiv installiert werden. Die neue Boot-Umgebung kann anschließend aktiviert werden, sodass sie zur aktuellen Boot-Umgebung wird.

Für die zulässigen Werte und Hintergrundinformationen zu diesem Schlüsselwort schlagen Sie bitte an folgenden Stellen nach:

- Eine Beschreibung der zulässigen Schlüsselwortwerte finden Sie unter „Profilschlüsselwörter und -werte“ auf Seite 116.
- Hintergrundinformationen zu Solaris Live Upgrade zum Erstellen, Aktualisieren und Aktivieren inaktiver Boot-Umgebungen finden Sie in Kapitel 2, „Solaris Live Upgrade (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.
- Hintergrundinformationen zur Arbeit mit einem Solaris Flash-Archiv finden Sie inKapitel 1, „Solaris Flash (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Flash-Archive (Erstellung und Installation)*.

BEISPIEL 3–12 Erstellen von RAID-1-Volumes bei der Installation eines Solaris Flash-Archivs

Im folgenden Beispiel gibt das Profil an, dass das benutzerdefinierte JumpStart-Programm mithilfe von Solaris Volume Manager jeweils RAID-1-Volumes (Mirrors) für die Dateisysteme Root (`/`), `swap`, `/usr` und `/export/home` erstellen soll. Ein Solaris Flash-Archiv wird in der Boot-Umgebung installiert.

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            flash_install
  arhcive_location        nfs server:/export/home/export/flash.s10.SUNWCall
  partitioning            explicit
  filesys                 mirror:d10 c0t0d0s0 c0t1d0s0 4096 /
  filesys                 mirror c0t0d0s1 2048 swap
  filesys                 mirror:d30 c0t0d0s3 c0t1d0s3 4096 /usr
  filesys                 mirror:d40 c0t0d0s4 c0t1d0s4 4096 /usr
  filesys                 mirror:d50 c0t0d0s5 c0t1d0s5 free /export/home
  metadb                  c0t1d0s7 size 8192 count 3
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

BEISPIEL 3–12 Erstellen von RAID-1-Volumes bei der Installation eines Solaris Flash-Archivs *(Fortsetzung)*

`install_type`	Das Profil installiert ein Solaris Flash-Archiv auf dem Klonsystem. Wie bei einer Erst- bzw. Neuinstallation werden alle Dateien überschrieben.
`archive_location`	Das Solaris Flash-Archiv wird von einem NFS-Server abgerufen.
`partitioning`	Mit dem Wert `explicit` legen Sie fest, dass die Dateisystem-Slices von den `filesys`-Schlüsselwörtern definiert werden.
`filesys`	Das Root-Dateisystem (`/`) wird auf den Slices `c0t1d0s0` und `c0t0d0s0` erstellt und gespiegelt. Das Root-Dateisystem (`/`) wird mit 4096 MB bemessen. Das RAID-1-Volume zur Spiegelung von `c0t0d0s0` und `c0t1d0s0` erhält den Namen `d10`.
`filesys`	Das `swap`-Dateisystem wird auf dem Slice `c0t0d0s1` erstellt und gespiegelt. Es wird mit 2048 MB bemessen. Das benutzerdefinierte JumpStart-Programm weist dem Mirror einen Namen zu.
`filesys`	Das `/usr`-Dateisystem wird auf den Slices `c0t1d0s3` und `c0t0d0s3` erstellt und gespiegelt. Das `/usr`-Dateisystem wird mit 4096 MB bemessen. Das RAID-1-Volume erhält den Namen `d30`.
`filesys`	Das `/usr`-Dateisystem wird auf den Slices `c0t1d0s4` und `c0t0d0s4` erstellt und gespiegelt. Das `/usr`-Dateisystem wird mit 4096 MB bemessen. Das RAID-1-Volume erhält den Namen `d40`.
`metadb`	Es werden drei auf 8192 Blöcke (4 MB) bemessene Statusdatenbankreplikationen (metadbs) auf Slice `c0t1d0s7` installiert.

- Einen Überblick zum Erstellen gespiegelter Dateisysteme während der Installation finden Sie in Kapitel 9, „Erstellen von RAID-1-Volumes (Mirrors) bei der Installation (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.
- Die Voraussetzungen und Richtlinien für die Erstellung gespiegelter Dateisysteme sind in Kapitel 10, „Erzeugen von RAID-1-Volumes (Mirrors) während der Installation (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades* beschrieben.
- Eine Beschreibung der Schlüsselwortwerte finden Sie unter „`filesys`-Profilschlüsselwort (Erstellen von RAID-1-Volumes)“ auf Seite 141 und „`metadb`-Profilschlüsselwort (Erstellen von Statusdatenbankreplikationen)“ auf Seite 149.

BEISPIEL 3–13 Erstellen eines RAID-1-Volumes zur Spiegelung des Root-Dateisystems

Das Profil im nachfolgenden Beispiel sieht vor, dass das benutzerdefinierte JumpStart-Programm unter Verwendung der Solaris Volume Manager-Technologie ein RAID-1-Volume (Mirror) für das Root-Dateisystem (`/`) erzeugt.

BEISPIEL 3–13 Erstellen eines RAID-1-Volumes zur Spiegelung des Root-Dateisystems *(Fortsetzung)*

```
# profile keywords        profile values
# ----------------        -------------------
  install_type            initial_install
  cluster                 SUNWCXall
  filesys                 mirror:d30 c0t1d0s0 c0t0d0s0  /
  filesys                 c0t0d0s3 512 swap
  metadb                  c0t0d0s4 size 8192 count 4
  metadb                  c0t1d0s4 size 8192 count 4
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`cluster` Die gesamte Solaris-Softwaregruppe plus OEM-Unterstützung, SUNWCXall, wird auf dem System installiert.

`filesys` Das Root-Dateisystem (`/`) wird auf den Slices `c0t1d0s0` und `c0t0d0s0` erstellt und gespiegelt. Das RAID-1-Volume zur Spiegelung von `c0t1d0s0` und `c0t0d0s0` erhält den Namen `d30`. Das benutzerdefinierte JumpStart-Programm weist den beiden Submirrors Namen zu.

`filesys` Das `swap`-Dateisystem wird auf dem Slice `c0t0d0s3` erstellt und gespiegelt. Es wird mit 512 MB bemessen.

`metadb` Es werden vier auf 8192 Blöcke (4 MB) bemessene Statusdatenbankreplikationen (metadbs) auf Slice `c0t0d0s4` installiert.

`metadb` Es werden vier auf 8192 Blöcke (4 MB) bemessene Statusdatenbankreplikationen (metadbs) auf Slice `c0t1d0s4` installiert.

- Einen Überblick zum Erstellen von RAID-1-Volumes während der Installation finden Sie in Kapitel 9, „Erstellen von RAID-1-Volumes (Mirrors) bei der Installation (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.
- Die Voraussetzungen und Richtlinien für die Erstellung von RAID-1-Volumes sind in Kapitel 10, „Erzeugen von RAID-1-Volumes (Mirrors) während der Installation (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades* aufgeführt.
- Eine Beschreibung der Schlüsselwortwerte finden Sie unter „`filesys`-Profilschlüsselwort (Erstellen von RAID-1-Volumes)“ auf Seite 141 und „`metadb`-Profilschlüsselwort (Erstellen von Statusdatenbankreplikationen)“ auf Seite 149.

BEISPIEL 3–14 Erstellen von RAID-1-Volumes zur Spiegelung mehrerer Dateisysteme

Das Profil im nachfolgenden Beispiel sieht vor, dass das benutzerdefinierte JumpStart-Programm unter Verwendung der Solaris Volume Manager-Technologie mehrere RAID-1-Volumes (Mirrors) zur Spiegelung des Root- (`/`), des `swap`- und des `/usr`-Dateisystems erzeugt.

```
# profile keywords        profile values
# ----------------        ------------------
  install_type            initial_install
  cluster                 SUNWCXall
  filesys                 mirror:d100 c0t1d0s0 c0t0d0s0 200 /
  filesys                 c0t1d0s5 500 /var
  filesys                 c0t0d0s5 500
  filesys                 mirror c0t0d0s1 512 swap
  metadb                  c0t0d0s3 size 8192 count 5
  filesys                 mirror c0t1d0s4 c0t0d0s4 2000 /usr
  filesys                 c0t1d0s7 free /export/home
  filesys                 c0t0d0s7 free
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`cluster` Die gesamte Solaris-Softwaregruppe plus OEM-Unterstützung, SUNWCXall, wird auf dem System installiert.

`filesys` Das Root-Dateisystem (`/`) wird auf den Slices `c0t1d0s0` und `c0t0d0s0` erstellt und gespiegelt. Das Root-Dateisystem (`/`) wird mit 200 MB bemessen. Das RAID-1-Volume zur Spiegelung von `c0t1d0s0` und `c0t0d0s0` erhält den Namen `d100`.

`filesys` Das Dateisystem `/var` wird auf Slice `c0t1d0s5` installiert und erhält die Kapazität 500 MB. Das Root-Dateisystem (`/`) wird auf den Slices `c0t1d0s0` und `c0t0d0s0` erstellt und gespiegelt. Das Root-Dateisystem (`/`) wird mit 200 MB bemessen. Das RAID-1-Volume zur Spiegelung von `c0t1d0s0` und `c0t0d0s0` erhält den Namen `d100`.

`filesys` Das `swap`-Dateisystem wird auf dem Slice `c0t0d0s1` erstellt und gespiegelt. Es wird mit 512 MB bemessen. Das benutzerdefinierte JumpStart-Programm weist dem Mirror einen Namen zu.

`metadb` Es werden fünf auf 8192 Blöcke (4 MB) bemessene Statusdatenbankreplikationen (metadbs) auf Slice `c0t0d0s3` installiert.

`filesys` Das `/usr`-Dateisystem wird auf den Slices `c0t1d0s4` und `c0t0d0s4` erstellt und gespiegelt. Das `/usr`-Dateisystem wird mit 2000 MB bemessen. Das benutzerdefinierte JumpStart-Programm weist dem Mirror einen Namen zu.

BEISPIEL 3–14 Erstellen von RAID-1-Volumes zur Spiegelung mehrerer Dateisysteme *(Fortsetzung)*

- Einen Überblick zum Erstellen gespiegelter Dateisysteme während der Installation finden Sie in Kapitel 9, „Erstellen von RAID-1-Volumes (Mirrors) bei der Installation (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.
- Die Voraussetzungen und Richtlinien für die Erstellung gespiegelter Dateisysteme sind in Kapitel 10, „Erzeugen von RAID-1-Volumes (Mirrors) während der Installation (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades* beschrieben.
- Eine Beschreibung der Schlüsselwortwerte finden Sie unter „`filesys`-Profilschlüsselwort (Erstellen von RAID-1-Volumes)“ auf Seite 141 und „`metadb`-Profilschlüsselwort (Erstellen von Statusdatenbankreplikationen)“ auf Seite 149.

BEISPIEL 3–15 x86: Verwenden des Schlüsselworts `fdisk`

```
# profile keywords      profile values
# ----------------      ------------------
  install_type          initial_install
  system_type           standalone

  fdisk                 c0t0d0 0x04 delete
  fdisk                 c0t0d0 solaris maxfree
  cluster               SUNWCall
  cluster               SUNWCacc delete
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`fdisk` Alle `fdisk`Partitionen des Typs DOSOS16 (04 hexadezimal) werden von der Festplatte `c0t0d0` gelöscht.

`fdisk` Eine Solaris-`fdisk`-Partition wird in dem größten zusammenhängenden freien Speicherbereich auf der Festplatte `c0t0d0` angelegt.

`cluster` Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert.

`cluster` Die Kontenführungsdienstprogramme, `SUNWCacc`, werden nicht auf dem System installiert.

Testen eines Profils

Nachdem Sie ein Profil erstellt haben, können Sie es mit dem Befehl `pfinstall(1M)` testen. Testen Sie das Profil, bevor Sie es für eine Installation oder ein Upgrade verwenden. Insbesondere beim Erstellen von Upgrade-Profilen, bei denen Festplattenplatz neu zugewiesen wird, ist das Testen sehr nützlich.

Anhand der von `pfinstall` generierten Ausgabe können Sie schnell ermitteln, ob das Profil wie beabsichtigt funktioniert. Sie können zum Beispiel ermitteln, ob ein System über ausreichend Festplattenspeicher für ein Upgrade auf ein neues Solaris-Release verfügt, bevor Sie das Upgrade ausführen.

Mit `pfinstall` können Sie ein Profil unter folgenden Bedingungen testen:

- Mit der Festplattenkonfiguration des Systems, auf dem `pfinstall` ausgeführt wird.
- Sonstige Festplattenkonfigurationen. Sie können mit einer Festplattenkonfigurationsdatei arbeiten, die die Struktur einer Festplatte angibt (z. B. Bytes/Sektor, Flags und Slices). Das Erstellen von Festplattenkonfigurationsdateien ist unter „Erstellen von Festplattenkonfigurationsdateien“ auf Seite 70 und „x86: So erstellen Sie eine Festplattenkonfigurationsdatei“ auf Seite 72 beschrieben.

Hinweis – Zum Testen eines Profils für ein Upgrade können Sie keine Festplattenkonfigurationsdatei verwenden. Sie müssen ein solches Profil mit der tatsächlichen Festplattenkonfiguration und der zurzeit auf dem System installierten Software testen.

▼ So erstellen Sie eine temporäre Solaris-Umgebung zum Testen eines Profils

Damit Sie ein Profil für ein bestimmtes Solaris-Release mit Erfolg korrekt testen können, müssen Sie das Profil innerhalb der Solaris-Umgebung desselben Release testen. Wenn Sie zum Beispiel ein Profil für eine Solaris-Neuinstallation testen möchten, müssen Sie den Befehl `pfinstall` auf einem System unter Solaris ausführen.

Sie müssen eine temporäre Installationsumgebung erstellen, wenn Sie ein Profil unter einer der folgenden Bedingungen testen wollen:

- Sie wollen ein Profil für ein Solaris 10 10/08-Upgrade auf einem System testen, auf dem eine frühere Version der Solaris-Software läuft.
- Sie haben noch kein Solaris 10 10/08-System zum Testen von Profilen für eine Solaris 10 10/08-Neuinstallation installiert.

1 Booten Sie ein System von einem Abbild einer der folgenden Datenträger:

Für SPARC-basierte Systeme:

- Solaris Operating System for SPARC Platforms-DVD
- Solaris Software for SPARC Platforms - 1-CD

Für x86-basierte Systeme:

- Solaris Operating System for x86 Platforms-DVD
- Solaris Software for x86 Platforms - 1-CD

Hinweis – Wenn Sie ein Upgrade-Profil testen möchten, booten Sie das System, auf dem das Upgrade ausgeführt werden soll.

2 **Beantworten Sie die Fragen zur Systemidentifikation.**

3 **Um das Installationsprogramm zu beenden, geben Sie ! an der folgenden Eingabeaufforderung ein.**

```
The Solaris installation program  will assist you in installing software for Solaris.
<Press ENTER to continue> {"!" exits}
```

4 **Führen Sie den Befehl `pfinstall` über die Shell aus. Weitere Informationen zum Befehl `pfinstall` finden Sie in Schritt 7 in „So testen Sie ein Profil“ auf Seite 54.**

▼ So testen Sie ein Profil

x86 nur – Wenn Sie das Schlüsselwort `locale` verwenden, schlägt der Befehl `pfinstall -D` zum Testen des Profils fehl. Die Beschreibung einer Abhilfe finden Sie in der Erläuterung zur Fehlermeldung „could not select locale,“ im Abschnitt „Upgrade von Solaris“ auf Seite 194.

1 **Verwenden Sie zum Testen des Profils ein System mit demselben Plattformtyp (SPARC bzw. x86) wie das System, für welches das Profil erstellt wurde.**

Wenn Sie ein Upgrade-Profil testen, müssen Sie das Profil auf dem System testen, auf dem das Upgrade ausgeführt werden soll.

2 **Entscheiden Sie anhand der folgenden Tabelle über das weitere Vorgehen.**

Testszenario	Anweisungen
Sie wollen ein Neuinstallationsprofil testen und verfügen über ein System, auf dem die Solaris 10 10/08-Software läuft.	Melden Sie sich bei dem System als Superuser an und fahren Sie mit Schritt 5 fort.
Sie wollen ein Upgrade-Profil testen oder Sie haben kein System mit Solaris 10 10/08, das zum Testen eines Neuinstallationsprofils verwendet werden kann.	Erstellen Sie zum Testen des Profils eine temporäre Solaris 10 10/08-Umgebung. Nähere Informationen finden Sie unter „So erstellen Sie eine temporäre Solaris-Umgebung zum Testen eines Profils“ auf Seite 53. Fahren Sie dann mit Schritt 3 fort.

3 Erstellen Sie einen temporären Einhängepunkt.

```
# mkdir /tmp/mnt
```

4 Hängen Sie das Verzeichnis mit den Profilen ein, die Sie testen wollen.

Einhängeszenario	Eingabeanweisungen
Einhängen eines entfernten NFS-Dateisystems für Systeme im Netzwerk	**mount -F nfs** *Servername*:*Pfad* **/tmp/mnt**
SPARC: Einhängen einer UFS-formatierten Diskette	**mount -F ufs /dev/diskette /tmp/mnt**
Einhängen einer PCFS-formatierten Diskette	**mount -F pcfs /dev/diskette /tmp/mnt**

5 Zum Testen des Profils mit einer bestimmten Hauptspeichergröße setzen Sie `SYS_MEMSIZE` auf die jeweilige Hauptspeichergröße in MB.

```
# SYS_MEMSIZE=Speicherkapazit[e4743c]fontreset
# export SYS_MEMSIZE
```

6 Haben Sie in Schritt 4 ein Verzeichnis eingehängt?

- Wenn ja, wechseln Sie in das Verzeichnis `/tmp/mnt`.

  ```
  # cd /tmp/mnt
  ```

- Wenn nicht, wechseln Sie in das Verzeichnis, in dem sich das Profil befindet. Dies ist normalerweise das JumpStart-Verzeichnis.

  ```
  # cd Jumpstart-Verz_pfad
  ```

7 Testen Sie das Profil mit dem Befehl `pfinstall`(1M).

```
# /usr/sbin/install.d/pfinstall -D:-d Plattenkonfigurationsdatei -c Pfad Profil
```

Achtung – Sie *müssen* die Option `-d` oder `-D` angeben. Wenn Sie nicht eine dieser Optionen angeben, verwendet `pfinstall` das angegebene Profil zum Installieren der Solaris-Software. Alle Daten auf dem System werden dabei überschrieben.

`-D`	`pfinstall` verwendet zum Testen des Profils die aktuelle Festplattenkonfiguration des Systems. Zum Testen eines Upgrade-Profils müssen Sie die Option `-D` verwenden.
`-d` *Plattenkonfigurationsdatei*	`pfinstall` verwendet zum Testen des Profils die Festplattenkonfigurationsdatei *Plattenkonfigurationsdatei.*

Wenn sich die *Plattenkonfigurationsdatei* nicht in dem Verzeichnis befindet, in dem `pfinstall` ausgeführt wird, müssen Sie den Pfad angeben.

Nähere Informationen zum Erstellen einer Festplattenkonfigurationsdatei finden Sie unter „Erstellen von Festplattenkonfigurationsdateien“ auf Seite 70.

Hinweis – Die Option `-d` *Plattenkonfigurationsdatei* können Sie bei einem Upgrade-Profil (`install_type upgrade`) nicht verwenden. Beim Testen eines Upgrade-Profils müssen Sie immer die Festplattenkonfiguration des Systems selbst verwenden, also die Option `-D`.

`-c` *Pfad*

Der Pfad zu dem Abbild der Solaris-Software. Diese Option verwenden Sie zum Beispiel, wenn auf dem System Volume Manager zum Einhängen der Solaris Software-1-CD für die jeweilige Plattform verwendet wird.

Hinweis – Die Option `-c` ist nicht erforderlich, wenn Sie von einem Abbild der Solaris Operating System-DVD oder der Solaris Software-1-CD für die jeweilige Plattform gebootet haben. Das DVD- oder CD-Abbild wird als Teil des Boot-Prozesses in `/cdrom/` eingehängt.

Profil

Der Name des zu testenden Profils. Wenn sich *Profil* nicht in dem Verzeichnis befindet, in dem `pfinstall` ausgeführt wird, müssen Sie den Pfad angeben.

Beispiele für das Testen von Profilen

Das folgende Beispiel zeigt, wie Sie mit `pfinstall` ein Profil mit dem Namen `basic_prof` testen können. Das Profil wird mit der Festplattenkonfiguration eines Systems getestet, auf dem die Solaris 10 10/08-Software installiert ist. Das Profil `basic_prof` befindet sich im Verzeichnis `/jumpstart`, und der Pfad zum Solaris Operating System-DVD-Abbild wird angegeben, da Volume Manager verwendet wird.

BEISPIEL 3–16 Profiltest mithilfe eines Solaris 10 10/08-Systems

```
# cd /jumpstart
# /usr/sbin/install.d/pfinstall -D -c /cdrom/Pfadname basic_prof
```

Das folgende Beispiel zeigt, wie Sie mit `pfinstall` ein Profil mit dem Namen `basic_prof` auf einem Solaris 10 10/08-System testen können. Der Test wird unter Verwendung der Festplattenkonfigurationsdatei `535_test` ausgeführt. Der Test prüft, ob 64 MB Hauptspeicher vorhanden sind. In diesem Beispiel wird ein Abbild der Solaris Software for SPARC Platforms - 1-CD bzw. der Solaris Software for x86 Platforms - 1-CD verwendet, das sich im Verzeichnis `/export/install` befindet.

BEISPIEL 3–17 Testen eines Profils mit einer Festplattenkonfigurationsdatei

```
# SYS_MEMSIZE=64
# export SYS_MEMSIZE
# /usr/sbin/install.d/pfinstall -d 535_test -c /export/install basic_prof
```

Validieren der Datei `rules`

Bevor Sie ein Profil und eine `rules`-Datei einsetzen können, müssen Sie mit dem Skript `check` überprüfen, ob die Dateien korrekt definiert sind. Wenn alle Regeln und Profile korrekt definiert sind, wird die Datei `rules.ok` erstellt, die von der benutzerdefinierten JumpStart-Installationssoftware benötigt wird, um ein System einem Profil zuzuordnen.

Tabelle 3–2 beschreibt die Vorgänge des `Check`-Skripts.

TABELLE 3–2 Was geschieht, wenn das Skript `check` ausgeführt wird?

Abschnitt	Beschreibung
1	Die Syntax der Datei `rules` wird überprüft. `check` überprüft, ob die Rule-Schlüsselwörter zulässig sind und ob für jede Regel die Felder *Beginn*, *Klasse* und *Ende* angegeben wurden. Die Felder *Beginn* und *Ende* können aus einem Minuszeichen (`-`) anstelle eines Dateinamens bestehen.
2	Wenn in der Datei `rules` keine Fehler gefunden werden, wird die Syntax aller in den Regeln angegebenen Profile überprüft.
3	Wenn keine Fehler gefunden werden, erstellt `check` die Datei `rules.ok` aus der Datei `rules`, entfernt alle Kommentare und Leerzeichen, behält alle Regeln bei und fügt die folgende Kommentarzeile am Ende an: `# version=2 checksum=num`

▼ So validieren Sie die Datei `rules`

1 **Stellen Sie sicher, dass sich das Skript `check` im JumpStart-Verzeichnis befindet.**

Hinweis – Das Skript `check` befindet sich im Verzeichnis `Solaris_10/Misc/jumpstart_sample` auf der Solaris Operating System-DVD bzw. der Solaris Software-1-CD.

---

2 **Wechseln Sie in das JumpStart-Verzeichnis.**

3 **Führen Sie das Skript `check` aus, um die `rules`-Datei zu validieren:**

```
$ ./check -p Pfad -r Dateiname
```

`-p` *Pfad* — Validiert die Datei `rules` unter Verwendung des Skripts `check` aus dem Abbild der Solaris-Software anstelle des Skripts `check` auf dem System, mit dem Sie arbeiten. *Pfad* ist der Pfad zu einem Abbild auf einer lokalen Festplatte oder zu einer eingehängten Solaris Operating System-DVD oder Solaris Software-1-CD.

Verwenden Sie diese Option, um die neueste Version von `check` auszuführen, wenn auf dem System eine frühere Version von Solaris läuft.

`-r` *Dateiname* — Gibt eine andere rules-Datei als die mit dem Namen `rules` an. Mit dieser Option können Sie die Gültigkeit einer Regel testen, bevor Sie die Regel in die Datei `rules` aufnehmen.

Während das Skript `check` ausgeführt wird, werden Meldungen zur Validierung der Datei `rules` und der einzelnen Profile ausgegeben. Wenn keine Fehler auftreten, gibt das Skript Folgendes aus:

```
The custom JumpStart configuration is ok
```

4 **Stellen Sie sicher, dass `root` Eigentümer der Datei `rules.ok` ist und dass die Berechtigungen auf 644 gesetzt sind.**

Siehe auch Nach der Validierung der Datei `rules` können Sie mehr zu optionalen Leistungsmerkmalen bei der benutzerdefinierten JumpStart-Installation in Kapitel 4, „Verwenden der optionalen Funktionen der benutzerdefinierten JumpStart-Installation (Vorgehen)“ nachlesen. Informationen zur Durchführung benutzerdefinierter JumpStart-Installationen finden Sie in in Kapitel 6, „Ausführen einer benutzerdefinierten JumpStart-Installation (Vorgehen)“.

◆ ◆ ◆ KAPITEL 4

Verwenden der optionalen Funktionen der benutzerdefinierten JumpStart-Installation (Vorgehen)

In diesem Kapitel werden die optionalen Funktionen beschrieben, die zum Erstellen zusätzlicher Tools für die benutzerdefinierte JumpStart-Installation zur Verfügung stehen.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „Erstellen von Begin-Skripten“ auf Seite 59
- „Erstellen von Finish-Skripten“ auf Seite 62
- „Erstellen einer komprimierten Konfigurationsdatei“ auf Seite 69
- „Erstellen von Festplattenkonfigurationsdateien“ auf Seite 70
- „Verwenden eines standortspezifischen Installationsprogramms“ auf Seite 76

Hinweis – Die Anweisungen in diesem Kapitel gelten für SPARC-Server oder x86-Server, auf denen benutzerdefinierte JumpStart-Dateien bereitgestellt werden, die so genannten Profilserver. Auf einem Profilserver können JumpStart-Dateien für unterschiedliche Plattformen zur Verfügung gestellt werden. So können Sie zum Beispiel auf einem SPARC-Server benutzerdefinierte JumpStart-Dateien für SPARC- und x86-Systeme bereitstellen.

Erstellen von Begin-Skripten

Ein Begin-Skript ist ein benutzerdefiniertes Bourne-Shell-Skript, das Sie in der Datei `rules` angeben. Ein Begin-Skript führt bestimmte Aufgaben aus, bevor die Solaris-Software auf einem System installiert wird. Sie können Begin-Skripte nur verwenden, wenn Sie die Solaris-Software mit dem benutzerdefinierten JumpStart-Installationsverfahren installieren.

Verwenden Sie ein Begin-Skript, um folgende Aufgaben auszuführen:

- Erstellen abgeleiteter Profile

- Sichern von Dateien vor einem Upgrade

Wichtige Informationen zu Begin-Skripten

- Geben Sie im Skript nichts an, was das Einhängen von Dateisystemen auf `/a` während einer Erst- oder Upgradeinstallation verhindern könnte. Wenn das JumpStart-Programm die Dateisysteme nicht in `/a` einhängen kann, tritt ein Fehler auf und die Installation schlägt fehl.
- Während der Installation wird die Ausgabe des Begin-Skripts in der Datei `/tmp/begin.log` gespeichert. Nach Abschluss der Installation wird die Protokolldatei wieder nach `/var/sadm/system/logs/begin.log` geleitet.
- Stellen Sie sicher, dass `root` Eigentümer des Begin-Skripts ist und die Berechtigungen auf 644 gesetzt sind.
- In Begin-Skripten können Sie benutzerdefinierte JumpStart-Umgebungsvariablen verwenden. Eine Liste der Umgebungsvariablen finden Sie unter „Benutzerdefinierte JumpStart-Umgebungsvariablen“ auf Seite 166.
- Speichern Sie Begin-Skripte im JumpStart-Verzeichnis.

Hinweis – Für die Release Solaris 10 wurde ein JumpStart-Beispielskript mit der Bezeichnung `set_nfs4_domain` auf einem Datenträger bereitgestellt, um eine Aufforderung während einer JumpStart-Installation zu verhindern. Das Skript unterdrückte die NFSv4-Eingabeaufforderung während der Installation. Dieses Skript ist nicht mehr erforderlich. **Ab Solaris-Release 10 10/08** wird das `sysidcfg`-Schlüsselwort `nfs4_domain` verwendet, das die Eingabeaufforderung unterdrückt. Das Skript `set_nfs4_domain` unterdrückt die Eingabeaufforderung nicht mehr.

Wenn auf Ihrem System nicht-globale Zonen installiert sind und das neue Schlüsselwort `nfs4_domain` in der Datei `sysidcfg` vorhanden ist, wird die Domäne durch das erste Booten einer nicht-globalen Zone eingestellt. Anderenfalls wird das interaktive Solaris-Installationsprogramm angezeigt, und Sie werden vor Abschluss des Boot-Vorgangs zur Eingabe eines Domänennamens aufgefordert.

Siehe „nfs4_domain-Schlüsselwort“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*

Erstellen abgeleiteter Profile mit einem Begin-Skript

Ein abgeleitetes Profil ist ein Profil, das bei einer benutzerdefinierten JumpStart-Installation von einem Begin-Skript dynamisch erstellt wird. Abgeleitete Profile benötigen Sie, wenn Sie die Datei `rules` nicht so einrichten können, dass bestimmte Systeme einem Profil zugeordnet

werden. So benötigen Sie eventuell abgeleitete Profile für identische Systemmodelle, die mit unterschiedlichen Hardwarekomponenten ausgestattet sind, also zum Beispiel für Systeme mit unterschiedlichen Grafikkarten.

Um eine Regel so einzurichten, dass ein abgeleitetes Profil verwendet wird, müssen Sie die folgenden Schritte ausführen:

- Setzen Sie das Profilfeld auf ein Gleichheitszeichen (=) statt eines Profils.
- Geben Sie im Begin-Feld ein Begin-Skript an, das ein abgeleitetes Profil für das System erstellt, auf dem Sie Solaris installieren wollen.

Wenn ein System einer Regel entspricht, die im Profilfeld ein Gleichheitszeichen (=) aufweist, erstellt das Begin-Skript das abgeleitete Profil, das zum Installieren der Solaris-Software auf dem System verwendet wird.

Das folgende Beispiel zeigt ein Begin-Skript, das jedes Mal dasselbe abgeleitete Profil erstellt. Sie können auch ein Begin-Skript schreiben, das je nach der Auswertung von Regeln unterschiedliche abgeleitete Profile erstellt.

BEISPIEL 4–1 Begin-Skript zum Erstellen eines abgeleiteten Profils

```
#!/bin/sh
echo "install_type        initial_install"    > ${SI_PROFILE}
echo "system_type         standalone"        >> ${SI_PROFILE}
echo "partitioning        default"           >> ${SI_PROFILE}
echo "cluster             SUNWCprog"         >> ${SI_PROFILE}
echo "package       SUNWman     delete"      >> ${SI_PROFILE}
echo "package       SUNWolman   delete"      >> ${SI_PROFILE}
echo "package       SUNWxwman   delete"      >> ${SI_PROFILE}
```

In diesem Beispiel muss das Begin-Skript den Namen des abgeleiteten Profils mithilfe der Umgebungsvariablen `SI_PROFILE` ermitteln, die standardmäßig auf `/tmp/install.input` gesetzt ist.

Hinweis – Wenn zum Erstellen eines abgeleiteten Profils ein Begin-Skript verwendet wird, stellen Sie sicher, dass das Skript keine Fehler aufweist. Ein abgeleitetes Profil lässt sich nicht mit dem Skript `check` verifizieren, denn abgeleitete Profile werden erst mit Ausführung des Begin-Skripts erstellt.

Erstellen von Finish-Skripten

Ein Finish-Skript ist ein benutzerdefiniertes Bourne-Shell-Skript, das Sie in der Datei `rules` angeben. Ein Finish-Skript führt bestimmte Aufgaben nach der Installation der Solaris-Software auf einem System auf, jedoch bevor das System erneut gebootet wird. Sie können Finish-Skripte nur verwenden, wenn Sie die Solaris-Software mit dem benutzerdefinierten JumpStart-Installationsverfahren installieren.

Mit einem Finish-Skript können Sie unter anderem die folgenden Aufgaben ausführen:

- Hinzufügen von Dateien
- Hinzufügen von einzelnen Packages oder Patches zusätzlich zu denen, die zusammen mit einer bestimmten Softwaregruppe installiert wurden
- Anpassen der Root-Umgebung
- Definieren des Root-Passworts für das System
- Installieren zusätzlicher Software

Wichtige Informationen zu Finish-Skripten

- Das Solaris-Installationsprogramm hängt die Dateisysteme des Systems auf `/a` ein. Die Dateisysteme bleiben bis zum Neustart des Systems in `/a` eingehängt. Mit einem Finish-Skript können Sie Dateien in der neu installierten Dateisystemhierarchie hinzufügen, ändern oder entfernen, indem Sie die in `/a` eingehängten Dateisysteme modifizieren.
- Während der Installation wird die Ausgabe des Finish-Skripts in der Datei `/tmp/finish.log` gespeichert. Nach Abschluss der Installation wird die Protokolldatei wieder nach `/var/sadm/system/logs/finish.log` geleitet.
- Stellen Sie sicher, dass `root` Eigentümer des Finish-Skripts ist und die Berechtigungen auf 644 gesetzt sind.
- In Finish-Skripten können Sie benutzerdefinierte JumpStart-Umgebungsvariablen verwenden. Eine Liste der Umgebungsvariablen finden Sie unter „Benutzerdefinierte JumpStart-Umgebungsvariablen“ auf Seite 166.
- Speichern Sie Finish-Skripte im JumpStart-Verzeichnis.

▼ So fügen Sie Dateien mit einem Finish-Skript hinzu

Mit einem Finish-Skript können Sie Dateien aus dem JumpStart-Verzeichnis zu einem bereits installierten System hinzufügen. Sie können Dateien hinzufügen, weil das JumpStart-Verzeichnis in dem Verzeichnis eingehängt ist, das in der Variablen `SI_CONFIG_DIR` angegeben wird. Standardmäßig ist dies das Verzeichnis `/tmp/install_config`.

Hinweis – Sie können Dateien auch ersetzen, indem Sie Dateien aus dem JumpStart-Verzeichnis über bereits vorhandene Dateien auf einem installierten System kopieren.

1. **Kopieren Sie alle Dateien, die Sie dem installierten System hinzufügen möchten, in das JumpStart-Verzeichnis.**

2. **Fügen Sie für jede in die neu installierte Dateisystemhierarchie zu kopierende Datei die folgende Zeile in das Finish-Skript ein:**

   `cp ${SI_CONFIG_DIR}/`*Dateiname* `/a/`*Pfadname*

Beispiel 4–2 Hinzufügen einer Datei mit einem Finish-Skript

Angenommen, es gibt eine speziell für alle Benutzer am Standort entwickelte Anwendung mit dem Namen `site_prog`. Wenn Sie eine Kopie von `site_prog` in das JumpStart-Verzeichnis stellen, bewirkt die folgende Zeile in einem Finish-Skript, dass `site_prog` aus dem JumpStart-Verzeichnis in das Verzeichnis `/usr/bin` eines Systems kopiert wird:

```
cp ${SI_CONFIG_DIR}/site_prog  /a/usr/bin
```

Hinzufügen von Packages oder Patches mit einem Finish-Skript

Sie können mit einem Finish-Skript nach der Installation der Solaris-Software automatisch Packages oder Patches hinzufügen. Indem Sie Packages mit einem Finish-Skript hinzufügen, sparen Sie Zeit und stellen außerdem sicher, dass Packages und Patches auf unterschiedlichen Systemen konsistent installiert werden.

Wenn Sie in Finish-Skripten den Befehl `pkgadd(1M)` oder `patchadd(1M)` verwenden, geben Sie mit der Option `-R` den Pfad `/a` als Root-Pfad an.

- Beispiel 4–3 zeigt ein Beispiel eines Finish-Skripts, das Packages hinzufügt.
- Beispiel 4–4 zeigt ein Beispiel eines Finish-Skripts, das Packages hinzufügt.

BEISPIEL 4–3 Hinzufügen von Packages mit einem Finish-Skript

```
#!/bin/sh

BASE=/a
MNT=/a/mnt
ADMIN_FILE=/a/tmp/admin

mkdir ${MNT}
mount -f nfs sherlock:/export/package ${MNT}
```

BEISPIEL 4–3 Hinzufügen von Packages mit einem Finish-Skript *(Fortsetzung)*

```
cat >${ADMIN_FILE} <<DONT_ASK
mail=root
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
DONT_ASK

/usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz
umount ${MNT}
rmdir ${MNT}
```

Nachfolgend werden einige Befehle dieses Beispiels erläutert.

- Der folgende Befehl hängt ein Verzeichnis auf einem Server ein, in dem sich das zu installierende Package befindet.

  ```
  mount -f nfs sherlock:/export/package ${MNT}
  ```

- Der folgende Befehl erzeugt die temporäre Package-Administrationsdatei `admin` und bewirkt, dass der Befehl `pkgadd(1M)` bei der Package-Installation keine Prüfung vornimmt und keine Eingabeaufforderungen ausgibt. Verwenden Sie die temporäre Package-Administrationsdatei, damit die Installation beim Hinzufügen von Packages ohne Benutzereingriff abläuft.

  ```
  cat >${ADMIN_FILE} <<DONT_ASK
  ```

- Der nachfolgende `pkgadd`-Befehl fügt das Package unter Verwendung der Option `-a` zur Angabe der Package-Administrationsdatei und der Option `-R` zur Angabe des Root-Pfads hinzu.

  ```
  /usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz
  ```

BEISPIEL 4–4 Hinzufügen von Patches mit einem Finish-Skript

```
 #!/bin/sh

########
#
# USER-CONFIGURABLE OPTIONS
```

BEISPIEL 4–4 Hinzufügen von Patches mit einem Finish-Skript *(Fortsetzung)*

```
#
########

# The location of the patches to add to the system after it's installed.
# The OS rev (5.x) and the architecture (`mach`) will be added to the
# root.  For example, /foo on a 8 SPARC would turn into /foo/5.8/sparc
LUPATCHHOST=ins3525-svr
LUPATCHPATHROOT=/export/solaris/patchdb
#########
#
# NO USER-SERVICEABLE PARTS PAST THIS POINT
#
#########

BASEDIR=/a

# Figure out the source and target OS versions
echo Determining OS revisions...
SRCREV=`uname -r`
echo Source $SRCREV

LUPATCHPATH=$LUPATCHPATHROOT/$SRCREV/`mach`

#
# Add the patches needed
#
echo Adding OS patches
mount $LUPATCHHOST:$LUPATCHPATH /mnt >/dev/null 2>&1
if [ $? = 0 ] ; then
    for patch in `cat /mnt/*Recommended/patch_order` ; do
        (cd /mnt/*Recommended/$patch ; echo yes | patchadd -u -d -R $BASEDIR .)
    done
    cd /tmp
    umount /mnt
else
    echo "No patches found"
if
```

Hinweis – Früher wurde der Befehl `chroot(1M)` zusammen mit dem Befehl `pkgadd` und `patchadd` in Finish-Skripten verwendet. In seltenen Fällen kann es vorkommen, dass bei einigen Packages oder Patches die Option `-R` nicht funktioniert. Sie müssen eine `/etc/mnttab`-Dummy-Datei im Root-Pfad `/a` erstellen, bevor Sie den Befehl `chroot` absetzen.

Zum Erstellen einer `/etc/mnttab`-Dummy-Datei fügen Sie die folgende Zeile zum Finish-Skript hinzu:

```
cp /etc/mnttab /a/etc/mnttab
```

Anpassen der Root-Umgebung mit einem Finish-Skript

Sie können mithilfe von Finish-Skripten auch Dateien anpassen, die bereits auf einem System installiert sind. Beispielsweise passt das Finish-Skript in Beispiel 4–5 die Root-Umgebung an, indem Informationen an die `.cshrc`-Datei im Root-Verzeichnis (`/`) angehängt werden.

BEISPIEL 4–5 Anpassen der Root-Umgebung mit einem Finish-Skript

```
#!/bin/sh
#
# Customize root's environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@‘uname -n‘> "
alias cp cp -i
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

Definieren des Root-Passworts eines Systems mit einem Finish-Skript

Nach der Installation der Solaris-Software wird das System neu gebootet. Vor dem Abschluss des Boot-Vorgangs fordert das System zur Eingabe des Root-Passworts auf. Der Boot-Vorgang kann erst abgeschlossen werden, wenn ein Passwort eingegeben wird.

Ein Finish-Skript mit dem Namen `set_root_pw` ist im Verzeichnis `auto_install_sample` gespeichert. Dieses Finish-Skript zeigt, wie das Root-Passwort automatisch, ohne Eingabeaufforderung definiert werden kann. `set_root_pw` wird in Beispiel 4–6 gezeigt .

Hinweis – Wenn Sie das Root-Passwort eines Systems mit einem Finish-Skript definieren, könnten Benutzer versuchen, das Root-Passwort anhand des verschlüsselten Passworts in dem Finish-Skript zu ermitteln. Sichern Sie das System gegen Versuche ab, das Root-Passwort auf diese Weise zu ermitteln.

BEISPIEL 4–6 Definieren des Root-Passworts eines Systems mit einem Finish-Skript

```
    #!/bin/sh
    #
    #       @(#)set_root_pw 1.4 93/12/23 SMI
    #
    # This is an example Bourne shell script to be run after installation.
    # It sets the system's root password to the entry defined in PASSWD.
    # The encrypted password is obtained from an existing root password entry
    # in /etc/shadow from an installed machine.

    echo "setting password for root"

    # set the root password
PASSWD=dKO5IBkSF42lw
    #create a temporary input file
cp /a/etc/shadow /a/etc/shadow.orig

    mv /a/etc/shadow /a/etc/shadow.orig
    nawk -F: '{
        if ( $1 == "root" )
          printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,passwd,$3,$4,$5,$6,$7,$8,$9
     else
               printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,$2,$3,$4,$5,$6,$7,$8,$9
     }' passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
#remove the temporary file
rm -f /a/etc/shadow.orig
# set the flag so sysidroot won't prompt for the root password
sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$
 mv /tmp/state.$$ ${SI_SYS_STATE}
```

Nachfolgend werden einige Befehle dieses Beispiels erläutert.

- Der folgende Befehl setzt die Variable `PASSWD` auf ein verschlüsseltes Root-Passwort, das aus einem vorhandenen Eintrag in der Datei `/etc/shadow` eines Systems abgerufen wird.

```
#create a temporary input file
```

BEISPIEL 4–6 Definieren des Root-Passworts eines Systems mit einem Finish-Skript *(Fortsetzung)*

- Der folgende Befehl erzeugt aus `/a/etc/shadow` eine temporäre Eingabedatei.

```
cp /a/etc/shadow /a/etc/shadow.orig
```

- Der folgende Befehl modifiziert den Root-Eintrag in der Datei `/etc/shadow` für das neu installierte System unter Verwendung von `$PASSWD` als Passwortfeld.

```
if ( $1 == "root" )
```

- Der folgende Befehl entfernt die temporäre Datei `/a/etc/shadow`.

```
rm -f /a/etc/shadow.orig
```

- Der folgende Befehl ändert den Eintrag in der state-Datei von `0` in `1` ab, so dass der Benutzer nicht zur Eingabe des Root-Passworts aufgefordert wird. Der Zugriff auf die state-Datei erfolgt über die Variable `SI_SYS_STATE`, die zurzeit den Wert `/a/etc/.sysIDtool.state` aufweist. Damit in den Skripten keine Probleme auftreten, wenn sich dieser Wert ändert, referenzieren Sie diese Datei immer mithilfe von `$SI_SYS_STATE`. Der hier gezeigte `sed`-Befehl enthält nach dem Zeichen `0` und nach dem Zeichen `1` ein Tabulatorzeichen.

```
sed -e 's/0 # root/1 # root/' ${SI_SYS_STATE} > /tmp/state.$$
```

Automatische Installationen mit Finish-Skripten

Sie können Finish-Skripten zur Installation zusätzlicher Software nach der Installation von Solaris einsetzen. Das Solaris-Installationsprogramm fordert Sie während der Installation zur Eingabe von Informationen auf. Damit die Installation ohne Benutzereingriffe abläuft, können Sie das Solaris-Installationsprogramm mit der Option `-nodisplay` oder `-noconsole` ausführen.

TABELLE 4–1 Solaris-Installationsoptionen

Option	Beschreibung
`-nodisplay`	Das Installationsprogramm wird ohne grafische Benutzeroberfläche ausgeführt. Verwenden Sie die Standardproduktinstallation, es sei denn, die Installation wurde mit der Option `-locales` modifiziert.
`-noconsole`	Die Installation wird ohne interaktives Text-Konsolengerät ausgeführt. Dies ist zusammen mit `-nodisplay` nützlich, wenn Sie UNIX-Skripte verwenden wollen.

Weitere Informationen entnehmen Sie bitte der Manpage `installer(1M)`.

Erstellen einer komprimierten Konfigurationsdatei

Anstatt den Speicherort der benutzerdefinierten JumpStart-Konfigurationsdateien mit dem Befehl `add_install_client` anzugeben, können Sie den Speicherort dieser Dateien auch beim Booten des Systems angeben. Sie können jedoch nur den Namen einer Datei angeben. Daher müssen Sie alle benutzerdefinierten JumpStart-Konfigurationsdateien in eine Datei komprimieren.

- **Bei SPARC-Systemen** geben Sie den Speicherort der Datei mit dem Befehl `boot` an.
- **Bei x86-Systemen** bearbeiten Sie zur Angabe des Speicherorts für die Datei den GRUB-Eintrag im GRUB-Menü.

Die komprimierte Konfigurationsdatei kann eines der folgenden Formate aufweisen:

- `tar`
- `tar` komprimiert
- `zip`
- `bzip tar`

▼ So erstellen Sie eine komprimierte Konfigurationsdatei

1 Wechseln Sie in das JumpStart-Verzeichnis auf dem Profilserver.

```
# cd Jumpstart-Verz_pfad
```

2 Komprimieren Sie die benutzerdefinierten JumpStart-Konfigurationsdateien mit einem Komprimierungstool in eine Datei.

Hinweis – Die komprimierte Konfigurationsdatei darf keine relativen Pfade enthalten. Die benutzerdefinierten JumpStart-Konfigurationsdateien müssen sich in demselben Verzeichnis wie die komprimierte Datei befinden.

Die komprimierte Konfigurationsdatei muss die folgenden Dateien enthalten:

- Profil
- `rules`
- `rules.ok`

Sie können auch die Datei `sysidcfg` in die komprimierte Konfigurationsdatei aufnehmen.

3 Speichern Sie die komprimierte Konfigurationsdatei auf einem NFS-Server, einem HTTP-Server oder auf einer lokalen Festplatte.

Beispiel für eine komprimierte Konfigurationsdatei

Das folgende Beispiel zeigt, wie Sie mit dem Befehl `tar` eine komprimierte Konfigurationsdatei mit dem Namen `config.tar` erstellen können. Die benutzerdefinierten JumpStart-Konfigurationsdateien befinden sich im Verzeichnis `/jumpstart`.

BEISPIEL 4–7 Erstellen einer komprimierten Konfigurationsdatei

```
# cd /jumpstart
# tar -cvf config.tar *
a profile 1K
a rules 1K
a rules.ok 1K
a sysidcfg 1K
```

Erstellen von Festplattenkonfigurationsdateien

In diesem Abschnitt wird beschrieben, wie Sie Konfigurationsdateien für einzelne und mehrere Festplatten erstellen können. Mit Festplattenkonfigurationsdateien können Sie `pfinstall(1M)` auf einem System zum Testen von Profilen für unterschiedliche Festplattenkonfigurationen einsetzen.

▼ SPARC: So erstellen Sie eine Festplattenkonfigurationsdatei

1 Suchen Sie ein SPARC-System mit einer zu testenden Festplatte.

2 Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.

Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

3 Erstellen einer Einzel–Festplattenkonfigurationsdatei durch das Umleiten der Ausgabe des Befehls `prtvtoc(1M)` an eine Datei.

`# prtvtoc /dev/rdsk/`*Gerätename* **`>`***Festplattenkonfigurationsdatei*

`/dev/rdsk/`*Gerätename*	Der Gerätename der Festplatte des Systems. *Gerätename* muss das Format `c`*w*`t`*x* `d`*y*`s2` oder `c` *x*`d`*y*`s2` aufweisen.
Festplattenkonfigurationsdatei	Der Name der Festplattenkonfigurationsdatei

4 Ermitteln Sie, ob Sie die Installation der Solaris-Software auf mehreren Festplatten testen müssen.

- Wenn nicht, Sie sind jetzt fertig.
- Wenn ja, verketten Sie die verschiedenen Einzelplatten-Konfigurationsdateien und speichern die Ausgabe in einer neuen Datei.

  ```
  # cat Festplattendatei1 Festplattendatei2 >Multiplatten-Konfigurationsdatei
  ```

  Die neue Datei wird zur Multiplatten-Konfigurationsdatei, wie im folgenden Beispiel gezeigt.

  ```
  # cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
  ```

5 Ermitteln Sie, ob die Zielnummern in den Festplattengerätenamen innerhalb der im vorherigen Schritt erstellten Multiplatten-Konfigurationsdatei eindeutig sind.

- Wenn ja, Sie sind jetzt fertig.
- Wenn nicht, öffnen Sie die Datei mit einem Texteditor und geben Sie eindeutige Zielnummern in die Festplattengerätenamen ein.

  Angenommen, die Datei enthält wie im folgenden Beispiel gezeigt dieselbe Zielnummer, `t0`, für verschiedene Festplattengerätenamen.

  ```
  * /dev/rdsk/c0t0d0s2 partition map
  ...
  * /dev/rdsk/c0t0d0s2 partition map
  ```

  Ändern Sie die zweite Zielnummer wie hier gezeigt in `t2`:

  ```
  * /dev/rdsk/c0t0d0s2 partition map
  ...
  * /dev/rdsk/c0t2d0s2 partition map
  ```

SPARC: Beispiel für eine Festplattenkonfigurationsdatei

Das folgende Beispiel zeigt, wie Sie eine Einzel–Festplattenkonfigurationsdatei, `104_test`, auf einem SPARC-basierten System mit einer 104-MB-Festplatte erstellen können.

BEISPIEL 4–8 SPARC: Erstellen einer Festplattenkonfigurationsdatei

Sie leiten die Ausgabe des Befehls `prtvtoc` in eine Einzelplatten-Konfigurationsdatei mit dem Namen `104_test` um:

```
# prtvtoc /dev/rdsk/c0t3d0s2 >104_test
```

BEISPIEL 4–8 SPARC: Erstellen einer Festplattenkonfigurationsdatei *(Fortsetzung)*

Der Inhalt der Datei `104_test` sieht etwa folgendermaßen aus:

```
* /dev/rdsk/c0t3d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      72 sectors/track
*      14 tracks/cylinder
*    1008 sectors/cylinder
*    2038 cylinders*    2036 accessible cylinders
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       1      2    00          0     164304   164303   /
       2      5    00          0    2052288  2052287
       3      0    00     164304     823536   987839   /disk2/b298
       5      0    00     987840     614880  1602719   /install/298/sparc/work
       7      0    00    1602720     449568  2052287   /space
```

Hier wurde beschrieben, wie Sie Festplattenkonfigurationsdateien für ein SPARC-basiertes System erstellen können. „Testen eines Profils“ auf Seite 52 enthält Informationen zum Verwenden von Festplattenkonfigurationsdateien zum Testen von Profilen.

▼ x86: So erstellen Sie eine Festplattenkonfigurationsdatei

1 Suchen Sie ein x86-basiertes System mit einer zu testenden Festplatte.

2 Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.

Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

3 Erstellen Sie einen Teil der Einzelplatten-Konfigurationsdatei, indem Sie die Ausgabe des Befehls `fdisk`(1M) in einer Datei speichern.

`# fdisk -R -W` *Festplattenkonfigurationsdatei* **`-h /dev/rdsk/`***Ger[e47465]name*

Festplattenkonfigurationsdatei Der Name der Festplattenkonfigurationsdatei.

`/dev/rdsk/`*Gerätename*	Der Gerätename des `fdisk`-Layouts der gesamten Festplatte. Der *Gerätename* muss das Format `c`*w*`t`*x*`d`*y*`s0` oder `c`*x*`d`*y*`s0` aufweisen.

4 Hängen Sie die Ausgabe des Befehls `prtvtoc`(1M) an die Festplattenkonfigurationsdatei an:

```
# prtvtoc /dev/rdsk/Gerätename >>Festplattenkonfiguration
```

`/dev/rdsk/`*Gerätename*	Der Gerätename der Festplatte des Systems. *Gerätename* muss das Format `c`*w*`t`*x* `d`*y*`s2` oder `c` *x*`d`*y*`s2` aufweisen.
Festplattenkonfig_datei	Der Name der Festplattenkonfigurationsdatei

5 Ermitteln Sie, ob Sie die Installation der Solaris-Software auf mehreren Festplatten testen müssen.

- Wenn nicht, Sie sind jetzt fertig.
- Wenn ja, verketten Sie die verschiedenen Einzelplatten-Konfigurationsdateien und speichern die Ausgabe in einer neuen Datei.

  ```
  # cat Festplattendatei1 Festplattendatei2 >Multiplatten-Konfiguration
  ```

  Die neue Datei wird zur Multiplatten-Konfigurationsdatei, wie im folgenden Beispiel gezeigt.

  ```
  # cat 104_disk2 104_disk3 104_disk5 >multi_disk_test
  ```

6 Ermitteln Sie, ob die Zielnummern in den Festplattengerätenamen innerhalb der im vorherigen Schritt erstellten Multiplatten-Konfigurationsdatei eindeutig sind.

- Wenn ja, Sie sind jetzt fertig.
- Wenn nicht, öffnen Sie die Datei mit einem Texteditor und geben eindeutige Zielnummern ein.

  Wenn die Datei wie im folgenden Beispiel gezeigt dieselbe Zielnummer, `t0`, für verschiedene Festplattengerätenamen enthält:

  ```
  * /dev/rdsk/c0t0d0s2 partition map
  ...
  * /dev/rdsk/c0t0d0s2 partition map
  ```

  Ändern Sie die zweite Zielnummer wie hier gezeigt in `t2`:

  ```
  * /dev/rdsk/c0t0d0s2 partition map
  ...
  * /dev/rdsk/c0t2d0s2 partition map
  ```

x86: Beispiel für eine Festplattenkonfigurationsdatei

Das folgende Beispiel zeigt, wie Sie eine Einzelplatten-Konfigurationsdatei, `500_test`, auf einem x86-basierten System mit einer 500-MB-Festplatte erstellen können.

BEISPIEL 4–9 x86: Erstellen einer Festplattenkonfigurationsdatei

Speichern Sie zunächst die Ausgabe des Befehls `fdisk` in einer Datei mit dem Namen `500_test`:

```
# fdisk -R -W 500_test -h /dev/rdsk/c0t0d0p0
```

Der Inhalt der Datei `500_test` ist unten aufgelistet:

```
 * /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
* 130:   SUNIXOS
*
* Id  Act Bhead Bsect   Bcyl  Ehead  Esect  Ecyl Rsect  Numsect
 130  128 44    3       0     46    30     1001 1410   2050140
```

Danach hängen Sie die Ausgabe des Befehls `prtvtoc` an die Datei `500_test` an:

```
# prtvtoc /dev/rdsk/c0t0d0s2 >>500_test
```

Die Datei `500_test` ist jetzt eine vollständige Festplattenkonfigurationsdatei:

BEISPIEL 4–9 x86: Erstellen einer Festplattenkonfigurationsdatei *(Fortsetzung)*

```
* /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
*  130:  SUNIXOS
*
* Id  Act Bhead Bsect Bcyl  Ehead  Esec  Ecyl Rsect  Numsect
 130  128 44    3     0     46    30    1001 1410   2050140
* /dev/rdsk/c0t0d0s2 partition map
*
* Dimensions:
*      512 bytes/sector
*       94 sectors/track
*       15 tracks/cylinder
*     1110 sectors/cylinder
*     1454 cylinders
*     1452 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*                          First    Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       2      5    01       1410   2045910   2047319
       7      6    00       4230   2043090   2047319  /space
       8      1    01          0      1410     1409
       9      9    01       1410      2820     422987
```

BEISPIEL 4–9 x86: Erstellen einer Festplattenkonfigurationsdatei *(Fortsetzung)*

Hier wurde beschrieben, wie Sie Festplattenkonfigurationsdateien für ein x86-basiertes System erstellen können. „Testen eines Profils“ auf Seite 52 enthält Informationen zum Verwenden von Festplattenkonfigurationsdateien zum Testen von Profilen.

Verwenden eines standortspezifischen Installationsprogramms

Sie können mithilfe von Begin- und Finish-Skripten auch ein eigenes Installationsprogramm zur Installation der Solaris-Software erstellen.

Wenn Sie im Profilfeld ein Minuszeichen (`-`) eingeben, wird die Installation von Solaris auf einem System von Begin- und Finish-Skripten und nicht über ein Profil und das Solaris-Installationsprogramm gesteuert.

Wenn zum Beispiel ein System der folgenden Regel entspricht, wird die Solaris-Software mit dem Begin-Skript `x_install.beg` und dem Finish-Skript `x_install.fin` auf dem System mit dem Namen `clover` installiert:

```
hostname clover x_install.beg - x_install.fin
```

◆ ◆ ◆ KAPITEL 5

Erstellen von benutzerdefinierten Rule- und Probe-Schlüsselwörtern (Vorgehen)

In diesem Kapitel finden Sie Informationen und Verfahren zum Erstellen eigener Rule- und Probe-Schlüsselwörter.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „Probe-Schlüsselwörter“ auf Seite 77
- „Erstellen einer `custom_probes`-Datei“ auf Seite 78
- „Validieren der Datei `custom_probes`“ auf Seite 81

Probe-Schlüsselwörter

Damit Sie verstehen können, was ein Probe-Schlüsselwort ist, müssen Sie zunächst wissen, was ein Rule-Schlüsselwort ist. Ein Rule-Schlüsselwort ist eine vordefinierte lexikalische Einheit oder ein Wort, die bzw. das ein allgemeines Systemattribut beschreibt, wie zum Beispiel den Host-Namen (`hostname`) oder die Hauptspeichergröße (`memsize`). Rule-Schlüsselwörter und die zugehörigen Werte ermöglichen es, ein System auf der Grundlage übereinstimmender Systemattribute einem Profil zuzuordnen. Das dem jeweiligen System zugeordnete Profil definiert, wie die Solaris-Software auf allen Systemen mit diesem Profil installiert werden soll.

Benutzerdefinierte JumpStart-Umgebungsvariablen, die in Begin- und Finish-Skripten eingesetzt werden, setzen Sie bei Bedarf. Informationen darüber, welches Betriebssystem bereits auf einem System installiert ist, stehen in `SI_INSTALLED` erst zur Verfügung, nachdem das Schlüsselwort `installed` verwendet wurde.

In manchen Situationen müssen Sie vielleicht dieselben Informationen aus einem Begin- oder Finish-Skript extrahieren, allerdings zu einem anderen Zweck als dem Zuordnen eines Systems und dem Ausführen eines Profils. Dieses Problem können Sie mit Probe-Schlüsselwörtern lösen. Probe-Schlüsselwörter dienen zum Extrahieren von Attributwerten, ohne dass Sie eine entsprechende Bedingung einrichten und ein Profil ausführen müssen.

Eine Liste der Schlüsselwörter und Werte finden Sie unter „Probe-Schlüsselwörter und -Werte“ auf Seite 168.

Erstellen einer custom_probes-Datei

Wenn die in „Rule-Schlüsselwörter und -Werte“ auf Seite 111 und „Probe-Schlüsselwörter und -Werte“ auf Seite 168 beschriebenen Rule- und Probe-Schlüsselwörter für Ihre Anforderungen nicht präzise genug sind, können Sie eigene Rule- bzw. Probe-Schlüsselwörter definieren, indem Sie eine custom_probes-Datei erstellen.

Die Datei custom_probes ist ein Bourne-Shell-Skript, das zwei Typen von Funktionen enthält. Sie müssen die Datei custom_probes in dem JumpStart-Verzeichnis speichern, in dem sich auch die Datei rules befindet. Sie können in einer custom_probes-Datei die folgenden zwei Funktionen definieren:

- Probe-Funktionen – Dienen zum Erfassen der benötigten Informationen oder zum Setzen einer entsprechenden SI_-Umgebungsvariablen, die Sie definieren. Probe-Funktionen werden zu Probe-Schlüsselwörtern.
- Vergleichsfunktionen – Rufen die entsprechende Probe-Funktion auf, vergleichen die Ausgabe der Probe-Funktion und geben 0 zurück, wenn das Schlüsselwort übereinstimmt, bzw. 1, wenn das Schlüsselwort nicht übereinstimmt. Comparison-Funktionen werden zu Rule-Schlüsselwörtern.

Syntax der Datei custom_probes

Die Datei custom_probes kann alle gültigen Befehle, Variablen und Algorithmen der Bourne-Shell enthalten.

Hinweis – Sie können Probe- und Comparison-Funktionen definieren, für die ein einzelnes Argument in der Datei custom_probes erforderlich ist. Wenn Sie das entsprechende benutzerdefinierte Probe-Schlüsselwort in der Datei rules verwenden, wird das Argument nach dem Schlüsselwort interpretiert (als $1).

Wenn Sie das entsprechende benutzerdefinierte Rule-Schlüsselwort in der Datei rules verwenden, wird das Argument so interpretiert, dass es nach dem Schlüsselwort beginnt und vor dem nächsten && oder Begin-Skript endet, je nachdem, was zuerst auftritt.

Die Datei custom_probes muss folgende Anforderungen erfüllen:

- Sie muss den Namen custom_probes aufweisen.
- Der Eigentümer muss root sein.
- Die Datei muss ausführbar und die Berechtigungen müssen auf 755 gesetzt sein.

- Die Datei muss mindestens eine Probe-Funktion und eine entsprechende Comparison-Funktion enthalten.

Um in einer solchen Datei eine klare Struktur zu erzielen, sollten Sie am Anfang der Datei zunächst alle Probe-Funktionen definieren und dann alle Comparison-Funktionen.

Syntax von Funktionsnamen in `custom_probes`

Der Name einer Probe-Funktion muss mit `probe_` anfangen. Der Name einer Comparison-Funktion muss mit `cmp_` anfangen.

Funktionen, die mit `probe_` anfangen, definieren neue Probe-Schlüsselwörter. Die Funktion `probe_tcx` definiert zum Beispiel das neue Probe-Schlüsselwort `tcx`. Funktionen, die mit `cmp_` anfangen, definieren neue Rule-Schlüsselwörter. `cmp_tcx` definiert zum Beispiel das neue Rule-Schlüsselwort `tcx`.

▼ So erstellen Sie eine `custom_probes`-Datei

1 **Erstellen Sie in einem Texteditor eine Textdatei für ein Bourne-Shell-Skript. Geben Sie der Datei den Namen `custom_probes`.**

2 **Definieren Sie in der Datei `custom_probes` die gewünschten Probe- und Comparison-Funktionen.**

Hinweis – Sie können Probe- und Comparison-Funktionen definieren, für die in der Datei `custom_probes` Argumente erforderlich sind. Wenn Sie das entsprechende benutzerdefinierte Probe-Schlüsselwort in der Datei `rules` verwenden, werden die Argumente nach dem Schlüsselwort nacheinander interpretiert (als $1, $2 usw.).

Wenn Sie das entsprechende benutzerdefinierte Rule-Schlüsselwort in der Datei `rules` verwenden, wird das Argument so interpretiert, dass es nach dem Schlüsselwort beginnt und vor dem nächsten && oder Begin-Skript endet, je nachdem, was zuerst auftritt.

---

3 **Speichern Sie die Datei `custom_probes` in dem JumpStart-Verzeichnis, das auch die Datei `rules` enthält.**

4 **Stellen Sie sicher, dass `root` Eigentümer der Datei `rules` ist und dass die Berechtigungen auf 644 gesetzt sind.**

Beispiele für custom_probes-Dateien und Schlüsselwörter

Zusätzliche Beispiele für Probe- und Comparison-Funktionen finden Sie in den folgenden Verzeichnissen:

- /usr/sbin/install.d/chkprobe auf einem System, auf dem die Solaris-Software installiert ist
- /Solaris_10/Tools/Boot/usr/sbin/install.d/chkprobe auf der Solaris Operating System-DVD oder der Solaris Software-1-CD

Die folgende custom_probes-Datei enthält eine Probe- und Comparison-Funktion, mit der geprüft wird, ob eine TCX-Grafikkarte vorhanden ist.

BEISPIEL 5–1 custom_probes-Datei

```
#!/bin/sh
#
# custom_probe script to test for the presence of a TCX graphics card.
#

#
# PROBE FUNCTIONS
#
probe_tcx() {
  SI_TCX=‘modinfo | grep tcx | nawk ’{print $6}’‘
  export SI_TCX
}

#
# COMPARISON FUNCTIONS
#
cmp_tcx() {
  probe_tcx

  if [ "X${SI_TCX}" = "X${1}" ]; then
     return 0
  else
     return 1
  if
}
```

Die folgende rules-Beispieldatei zeigt, wie Sie das im vorherigen Beispiel definierte Probe-Schlüsselwort, tcx, verwenden können. Wenn eine TCX-Grafikkarte installiert ist und erkannt wird, wird das Profil profile_tcx ausgeführt. Andernfalls wird das Profil profile ausgeführt.

Hinweis – Stellen Sie Probe-Schlüsselwörter immer möglichst an den Anfang der Datei `rules`. So stellen Sie sicher, dass die Schlüsselwörter vor Rule-Schlüsselwörtern gelesen und ausgeführt werden, die von den Probe-Schlüsselwörtern abhängen.

BEISPIEL 5–2 Benutzerdefiniertes Probe-Schlüsselwort in einer `rules`-Datei

```
probe tcx
tcx     tcx     -     profile_tcx     -
any     any     -     profile         -
```

Validieren der Datei `custom_probes`

Bevor Sie ein Profil, eine `rules`- und eine `custom_probes`-Datei einsetzen können, müssen Sie mit dem `check`-Skript überprüfen, ob die Dateien korrekt definiert sind. Wenn alle Profile, Regeln, Probe- und Comparison-Funktionen korrekt definiert sind, werden die Datei `rules.ok` und die Datei `custom_probes.ok` erstellt. Tabelle 5–1 beschreibt den Ablauf des Skripts `check`.

TABELLE 5–1 Was geschieht, wenn das Skript `check` ausgeführt wird?

Abschnitt	Beschreibung
1	`check` sucht eine `custom_probes`-Datei.
2	Wenn die Datei vorhanden ist, erstellt `check` die Datei `custom_probes.ok` aus der Datei `custom_probes`, entfernt alle Kommentare und Leerzeilen und behält alle Bourne-Shell-Befehle, Variablen und Algorithmen bei. Danach fügt `check` die folgende Kommentarzeile an das Ende der Datei an: `# version=2 checksum=num`

▼ So validieren Sie die Datei `custom_probes`

1. **Stellen Sie sicher, dass sich das Skript `check` im JumpStart-Verzeichnis befindet.**

   Hinweis – Das Skript `check` befindet sich im Verzeichnis `Solaris_10/Misc/jumpstart_sample` auf der Solaris Operating System-DVD bzw. der Solaris Software-1-CD.

2. **Wechseln Sie in das JumpStart-Verzeichnis.**

3. **Führen Sie das Skript `check` aus, um die Datei `rules` und die Datei `custom_probes` zu validieren.**

   `$ ./check -p` *Pfad* `-r` *Dateiname*

-p *Pfad*	Validiert die Datei custom_probes unter Verwendung des Skripts check aus dem Abbild der Solaris-Software für die jeweilige Plattform anstelle des Skripts check auf dem System, mit dem Sie arbeiten. *Pfad* ist der Pfad zu einem Abbild auf einer lokalen Festplatte oder zu einer eingehängten Solaris Operating System-DVD oder Solaris Software-1-CD. Verwenden Sie diese Option, um die neueste Version von check auszuführen, wenn auf dem System eine frühere Version von Solaris ausgeführt wird.
-r *Dateiname*	Gibt eine andere Datei als die mit dem Namen custom_probes an. Mithilfe der Option -r können Sie die Gültigkeit einer Reihe von Funktionen prüfen, bevor Sie die Funktionen in die Datei custom_probes aufnehmen.

Bei der Ausführung des check-Skripts macht das Skript Angaben zur Gültigkeit der Dateien rules und custom_probes sowie jedes einzelnen Profils. Wenn keine Fehler auftreten, gibt das Skript Folgendes aus: „The custom JumpStart configuration is ok“ . Im JumpStart-Verzeichnis werden die Dateien rules.ok und custom_probes.ok erstellt.

4 Ermitteln Sie, ob die Datei custom_probes.ok ausführbar ist.

- Wenn ja, fahren Sie mit Schritt 5 fort.
- Wenn nicht, geben Sie den folgenden Befehl ein:

```
# chmod +x custom_probes
```

5 Stellen Sie sicher, dass root Eigentümer der Datei custom_probes.ok ist und dass die Berechtigungen auf 755 gesetzt sind.

◆ ◆ ◆ KAPITEL 6

Ausführen einer benutzerdefinierten JumpStart-Installation (Vorgehen)

In diesem Kapitel wird beschrieben, wie Sie eine benutzerdefinierte JumpStart-Installation auf einem SPARC- oder x86-basierten System ausführen. Sie müssen diese Verfahren auf dem System ausführen, auf dem die Solaris-Software installiert werden soll.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus“ auf Seite 89
- „x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus“ auf Seite 94

Einschränkungen für eine JumpStart-Installation

Während einer JumpStart-Installation können eine Reihe von Problemen auftreten. Genauere Informationen können Sie der folgenden Tabelle entnehmen.

TABELLE 6–1 Einschränkungen bei JumpStart-Installationen

Problem	Beschreibung	Weitere Informationen
Das JumpStart-Beispielskript ist nicht mehr erforderlich, um die NFSv4-Eingabeaufforderung zu unterdrücken	Für die Release Solaris 10 wurde ein JumpStart-Beispielskript mit der Bezeichnung `set_nfs4_domain` auf einem Datenträger bereitgestellt, um eine Aufforderung während einer JumpStart-Installation zu verhindern. Das Skript unterdrückte die NFSv4-Eingabeaufforderung während der Installation. Dieses Skript ist nicht mehr erforderlich. **Ab der Solaris Release 10 8/07** wird das `sysidcfg`-Schlüsselwort `nfs4_domain` verwendet, das die Eingabeaufforderung unterdrückt. Das Skript `set_nfs4_domain` unterdrückt die Eingabeaufforderung nicht mehr. Wenn auf Ihrem System nicht-globale Zonen installiert sind und das neue Schlüsselwort `nfs4_domain` in der Datei `sysidcfg` vorhanden ist, wird die Domäne durch das erste Booten einer nicht-globalen Zone eingestellt. Anderenfalls wird das interaktive Solaris-Installationsprogramm angezeigt, und Sie werden vor Abschluss des Boot-Vorgangs zur Eingabe eines Domänennamens aufgefordert.	„nfs4_domain-Schlüsselwort“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
Auswahl einer Tastatursprache in der Datei `sysidcfg` verhindert eine Eingabeaufforderung	Wenn sich die Tastatur nicht selbst identifiziert und Sie verhindern möchten, dass die Eingabeaufforderung während der JumpStart-Installationsmethode angezeigt wird, müssen Sie die Tastatursprache in der Datei `sysidcfg` auswählen. Bei der JumpStart-Installationsmethode wird standardmäßig die Sprache U.S. Englisch installiert. Legen Sie das Tastatur-Schlüsselwort in der Datei `sysidcfg` fest, um eine andere Sprache und das entsprechende Tastaturlayout auszuwählen.	■ „Schlüsselwörter in der Datei sysidcfg“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation* ■ Lesen Sie auch die folgenden Manpages: ■ `sysidtool(1M)` ■ `sysidcfg(4)`
Falls auf Ihrem System bereits nicht-globale Zonen installiert sind, verwenden Sie zum Aktualisieren des Systems Solaris Live Upgrade	Sie können ein System mit nicht-globalen Zonen auch mit JumpStart aktualisieren, wir empfehlen Ihren jedoch hierzu Solaris Live Upgrade. JumpStart benötigt eventuell deutlich mehr Zeit, da die für die Aktualisierung erforderliche Zeit linear mit der Anzahl an installierten nicht-globalen Zonen ansteigt.	*Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*
Ein Solaris Flash-Archiv kann keine nicht-globalen Zonen enthalten.	Wenn Sie ein Solaris Flash-Archiv zur Installation verwenden, wird ein Archiv mit nicht-globalen Zonen nicht korrekt in Ihrem System installiert.	Allgemeine Informationen zum Erstellen von nicht-globalen Zonen finden Sie unter *Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones*.

TABELLE 6–1 Einschränkungen bei JumpStart-Installationen *(Fortsetzung)*

Problem	Beschreibung	Weitere Informationen
Ein Solaris Flash-Archiv kann nur von einem UFS-Dateisystem installiert werden.	Sie können ein Solaris Flash-Archiv nicht von einem ZFS-Root-Pool erstellen. Sie können ein Solaris Flash-Archiv nicht auf einem ZFS-Root-Pool installieren.	
SPARC: Zusätzliche Hardware-Anforderungen	Bitte prüfen Sie Ihre Hardware-Dokumentation auf zusätzliche Anforderungen für Ihre Plattform, die eventuell für die Verwendung der JumpStart-Installationsmethode erforderlich sind.	

SPARC: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation

TABELLE 6–2 Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation

Aufgabe	Beschreibung	Siehe
Überprüfen Sie, ob das System unterstützt wird.	Informationen zur Systemunterstützung in der Solaris-Umgebung finden Sie in der Hardwaredokumentation.	*Solaris Handbuch zur Hardware-Plattform von Sun* unter `http://docs.sun.com`
Überprüfen Sie, ob das System über genügend Festplattenspeicher für die Solaris-Software verfügt.	Überprüfen Sie, ob auf dem System ausreichend Festplattenspeicher für die Installation der Solaris-Software vorhanden ist.	Kapitel 4, „Systemvoraussetzungen, Richtlinien und Upgrades (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
(Optional) Legen Sie die Systemparameter fest.	Sie können die Systeminformationen vorkonfigurieren und so vermeiden, dass Sie während des Installations- bzw. Upgrade-Vorgangs dazu aufgefordert werden, diese Informationen einzugeben.	Kapitel 2, „Vorkonfigurieren der Systemkonfigurationsinformationen (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
Bereiten Sie das System auf die benutzerdefinierte JumpStart-Installation vor.	Erstellen und validieren Sie eine `rules`-Datei und Profildateien.	Kapitel 3, „Vorbereiten von benutzerdefinierten JumpStart-Installationen (Vorgehen)“

TABELLE 6–2 Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation *(Fortsetzung)*

Aufgabe	Beschreibung	Siehe
(Optional) Bereiten Sie optionale benutzerdefinierte JumpStart-Funktionen vor.	Wenn Sie begin-Skripten, finish-Skripten oder andere optionale Funktionen nutzen wollen, bereiten Sie die Skripten bzw. Dateien vor.	Kapitel 4, „Verwenden der optionalen Funktionen der benutzerdefinierten JumpStart-Installation (Vorgehen)“ und Kapitel 5, „Erstellen von benutzerdefinierten Rule- und Probe-Schlüsselwörtern (Vorgehen)“
(Optional) Bereiten Sie die Installation der Solaris-Software über das Netzwerk vor.	Wenn Sie ein System von einem Remote-Abbild der Solaris Operating System-DVD oder der Solaris Software for SPARC Platforms-CD installieren wollen, müssen Sie das System so einrichten, dass es von einem Installations- oder einem Boot-Server aus gebootet und installiert werden kann.	Kapitel 5, „Installieren über das Netzwerk mithilfe von DVDs (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation* Kapitel 6, „Installieren über das Netzwerk mithilfe von CDs (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
(Optional) Bereiten Sie die Solaris Flash-Archivinstallation vor.	Vor der Installation eines Solaris Flash-Archivs sind bestimmte Schritte durchzuführen.	„So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor“ auf Seite 87
Führen Sie die Installation oder das Upgrade aus.	Booten Sie das System, um die Installation bzw. das Upgrade zu initiieren.	„SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus“ auf Seite 89

SPARC: Ausführen einer benutzerdefinierten JumpStart-Installation

Während einer benutzerdefinierten JumpStart-Installation versucht das JumpStart-Programm, das zu installierende System den Regeln in der Datei `rules.ok` zuzuordnen. Das JumpStart-Programm liest die Regeln nacheinander. Eine Übereinstimmung liegt vor, wenn das zu installierende System alle in der Regel definierten Systemattribute aufweist. Sobald eine Regel gefunden wird, die dem System entspricht, liest das JumpStart-Programm die Datei `rules.ok` nicht weiter und beginnt mit der Installation des Systems auf der Grundlage des in der Regel angegebenen Profils.

▼ So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor

Sie können bei der Installation ein vollständiges Archiv für eine Erstinstallation oder ein Differenzarchiv für die Aktualisierung eines bereits installierten Archivs verwenden. Sie können die benutzerdefinierte JumpStart-Installation oder Solaris Live Upgrade verwenden, um ein Archiv in eine inaktive Boot-Umgebung zu installieren. Das nachfolgende Verfahren beschreibt die Installation eines Archivs mit der benutzerdefinierten JumpStart-Installation.

- Einen Überblick zu vollständigen und Differenzarchiven finden Sie in Kapitel 1, „Solaris Flash (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Flash-Archive (Erstellung und Installation)*.
- Anleitungen zum Installieren eines Archivs in einer inaktiven Boot-Umgebung mithilfe von Solaris Live Upgrade finden Sie unter „So installieren Sie ein Solaris Flash-Archiv mit einem Profil“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

1 Beachten Sie die folgenden Einschränkungen.

Beschreibung	Beispiel
Vorsicht: Wenn Sie das Schlüsselwort `archive_location` zur Installation eines Solaris Flash-Archivs verwenden, müssen das Archiv und die Installationsdatenträger identische Betriebssystemversionen enthalten.	Wenn es sich beim Archiv beispielsweise um Solaris 10 10/08 handelt und Sie DVDs verwenden, müssen Sie das Archiv mit der Solaris 10 10/08 -DVD installieren. Sollten Sie stattdessen unterschiedliche Betriebssystemversionen verwenden, so schlägt die Installation auf dem Klonsystem fehl.
Achtung – Solaris Flash-Archive können nicht korrekt erstellt werden, wenn eine nicht-globale Zone installiert ist. Die Solaris Flash-Funktion ist nicht mit der Solaris Zones-Partitionierungstechnologie kompatibel. Wenn Sie ein Solaris Flash-Archiv erstellen, wird dieses Archiv nicht korrekt installiert, wenn es unter den folgenden Bedingungen bereitgestellt wird: ■ Das Archiv wird in einer nicht-globalen Zone erstellt. ■ Das Archiv wird in einer globalen Zone erstellt, in der nicht-globale Zonen installiert sind.	

2 Legen Sie auf dem Installationsserver die benutzerdefinierte JumpStart-Datei `rules` an.

Ausführliche Anweisungen zum Erstellen benutzerdefinierter JumpStart-Dateien finden Sie in Kapitel 3, „Vorbereiten von benutzerdefinierten JumpStart-Installationen (Vorgehen)“.

3 Legen Sie auf dem Installationsserver die benutzerdefinierte JumpStart-Profildatei an.

Beispiele für Solaris Flash-Archivprofile finden Sie unter „Beispiele für Profile“ auf Seite 40.

Bei der Installation eines Solaris Flash-Archivs sind aus der bestehenden Liste von benutzerdefinierten JumpStart-Schlüsselwörtern in Tabelle 8–2 ausschließlich die folgenden Schlüsselwörter zulässig:

Schlüsselwort	Erste Installation	Differenzarchiv
(Obligatorisch)`archive_location`	X	X
`fdisk` (nur x86)	X	X
`filesys`	X	
Hinweis – Es ist nicht möglich, das Schlüsselwort `filesys` auf den Wert `auto` zu setzen.		
`forced_deployment`		X
(erforderlich) `install_type`	X	X
`local_customization`	X	X
`no_content_check`		X
`no_master_check`		X
`Package`	X	
`root_device`	X	X

a. Setzen Sie den Wert des Schlüsselworts `install_type` auf einen der nachfolgenden Typen.

- Für vollständige Archivinstallationen setzen Sie den Wert auf `flash_install`
- Für Installationen von Differenzarchiven setzen Sie den Wert auf `flash_update`

b. Fügen Sie den Pfad zum Solaris Flash-Archiv über das Schlüsselwort `archive_location` hinzu.

Genaue Angaben zum Schlüsselwort `archive_location` finden Sie im Abschnitt „`archive_location`-Schlüsselwort“ auf Seite 118.

c. **Geben Sie die Dateisystemkonfiguration an.**

   Das automatische Partitions-Layout wird bei der Extraktion des Solaris Flash-Archivs nicht unterstützt.

d. **(Optional) Wenn Sie bei der Archiv-Installation gleichzeitig noch zusätzliche Packages installieren möchten, verwenden Sie hierzu das Schlüsselwort `package`. Weitere Informationen finden Sie unter „`package`-Profilschlüsselwort (UFS und ZFS)“ auf Seite 151.**

e. **(Optional) Wenn Sie auf dem Klonsystem zusätzliche Solaris Flash-Archive installieren möchten, fügen Sie für jedes zu installierende Archiv eine `archive_location`-Zeile hinzu.**

4 **Fügen Sie auf dem Installationsserver die Clients hinzu, die Sie mit dem Solaris Flash-Archiv installieren möchten.**

Ausführliche Anweisungen finden Sie unter:

- „Hinzufügen der über das Netzwerk zu installierenden Systeme mit einem DVD-Abbild“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
- „Hinzufügen der über das Netzwerk zu installierenden Systeme mit einem CD-Abbild“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*

5 **Nehmen Sie die benutzerdefinierte JumpStart-Installation auf den Klonsystemen vor.**

Ausführliche Anweisungen finden Sie unter „SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus“ auf Seite 89.

▼ SPARC: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm aus

1 **Wenn das System Teil eines Netzwerks ist, stellen Sie sicher, dass ein Ethernet-Anschluss oder ein ähnlicher Netzwerkadapter mit dem System verbunden ist.**

2 **Wenn Sie ein System über eine `tip(1)`-Verbindung installieren wollen, muss das Fenster mindestens 80 Spalten breit und 24 Zeilen hoch sein.**

Die aktuelle Größe des `tip`-Fensters können Sie mit dem Befehl `stty(1)` ermitteln.

3 **Wenn Sie zur Installation der Solaris-Software das DVD-ROM- oder CD-ROM-Laufwerk des Systems verwenden, legen Sie die Solaris Operating System for SPARC Platforms-DVD oder die Solaris Software for SPARC Platforms - 1-CD in das Laufwerk ein.**

4 **Wenn Sie eine Profildiskette verwenden, legen Sie die Profildiskette in das Diskettenlaufwerk des Systems ein.**

5 Booten Sie das System.

- Bei einem neuen System schalten Sie dieses zunächst ein.
- Wenn Sie eine Installation bzw. ein Upgrade auf einem vorhandenen System ausführen wollen, fahren Sie das System herunter. Geben Sie an der Eingabeaufforderung `ok` die geeigneten Optionen für den boot-Befehl ein. Für den Befehl `boot` gilt folgende Syntax:

  ```
  ok boot [CD–DVD|net] - install [URL|ask] Optionen
  ```

  Wenn Sie beispielsweise folgenden Befehl eingeben, wird das BS mithilfe eines JumpStart-Profils über das Netzwerk installiert.

  ```
  ok boot net - install http://131.141.2.32/jumpstart/config.tar
  ```

  In der nachfolgenden Tabelle sind die Optionen des Boot-Befehls beschrieben.

SPARC nur – Das System prüft die Hardware- und Systemkomponenten und das SPARC-System wird gebootet. Der Boot-Vorgang dauert mehrere Minuten.

---

6 Wenn Sie die Systeminformationen nicht in der Datei `sysidcfg` vorkonfiguriert haben, beantworten Sie die Fragen zur Systemkonfiguration.

7 Befolgen Sie die Anweisungen auf dem Bildschirm und installieren Sie die Software.

Nach Abschluss der Solaris-Software durch das JumpStart-Programm wird das System automatisch neu gestartet.

Nach Abschluss der Installation werden die Installationsprotokolle in einer Datei gespeichert. Die Installationsprotokolle finden Sie in den folgenden Verzeichnissen:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

SPARC: Befehlsreferenz für den Befehl `boot`

Für den Befehl `boot` gilt folgende Syntax:

```
ok boot [CD–DVD|net] - install [URL|ask] Optionen
```

In der folgenden Tabelle sind die für eine JumpStart-Installation geeigneten Befehlszeilenoptionen des Befehls `boot` beschrieben.

Option	Beschreibung
[*CD–DVD*|net]	Gibt an, ob von einer CD, einer DVD oder einem Installationsserver im Netzwerk gebootet wird. ■ *CD-DVD* - Geben Sie zum Booten von einer CD oder DVD `cdrom` an. ■ `net` - Gibt an, dass von einem Installationsserver im Netzwerk gebootet werden soll.

[*URL*| ask]	Gibt die Adresse der benutzerdefinierten JumpStart-Dateien an oder fordert zu deren Eingabe auf.
	■ *URL* – Gibt den Pfad zu den Dateien an. Sie können einen URL für Dateien angeben, die auf einem HTTP- oder HTTPS-Server abgelegt sind.
	HTTP-Server
	`http://`*Servername:IP_address/jumpstart_dir_path/Komprimierte_Konfigurationsdatei&Proxy_Info*
	■ Wenn Sie eine `sysidcfg`-Datei in die komprimierte Konfigurationsdatei aufgenommen haben, müssen Sie wie im folgenden Beispiel die IP-Adresse des Servers angeben, auf dem sich die Datei befindet:
	`http://131.141.2.32/jumpstart/config.tar`
	■ Wenn Sie die komprimierte Konfigurationsdatei auf einem HTTP-Server hinter einer Firewall gespeichert haben, müssen Sie während des Boot-Vorgangs einen Proxy-Server angeben. Sie brauchen keine IP-Adresse für den Server anzugeben, auf dem sich die Datei befindet. Sie müssen jedoch wie im folgenden Beispiel eine IP-Adresse für den Proxy-Server angeben:
	`http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151`
	■ `ask` – Legt fest, dass Sie das Installationsprogramm zur Eingabe der Adresse der komprimierten Konfigurationsdatei auffordert. Die Aufforderung erscheint, nachdem das System gestartet und die Verbindung zum Netzwerk hergestellt wurde. Bei Verwendung dieser Option können Sie keine vollständig automatische JumpStart-Installation durchführen.
	Wenn Sie durch Drücken der Eingabetaste die Eingabeaufforderung umgehen, konfiguriert das Solaris-Installationsprogramm die Netzwerkparameter interaktiv. Danach fordert Sie das Installationsprogramm zur Eingabe des Speicherorts der komprimierten Konfigurationsdatei auf.

Optionen	■ `dhcp` – Legt fest, dass die zum Booten des Systems erforderlichen Netzwerkinformationen über einen DHCP-Server abgerufen werden. Diese Option wird bei einer JumpStart-Installation nicht benötigt. Wenn Sie `dhcp` weglassen und somit angeben, dass kein DHCP-Server verwendet werden soll, verwendet das System die Datei `/etc/bootparams` oder die Datenbank `bootparams` des Naming Service. Sie würden zum Beispiel nicht `dhcp` angeben, wenn Sie eine statische IP-Adresse beibehalten wollen. ■ Die Optionen `nowin` und `text` sind für JumpStart-Installationen irrelevant. Sie sind bei interaktiven Installationen bedeutsam. Weitere Informationen finden Sie unter „So führen Sie eine Installation bzw. ein Upgrade mit dem Solaris-Installationsprogramm aus“ in *Solaris 10 10/08 Installationshandbuch: Grundinstallationen*.

x86: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation

TABELLE 6–3 x86: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation

Aufgabe	Beschreibung	Siehe
Ermitteln Sie, ob ein vorhandenes Betriebssystem und Benutzerdaten beibehalten werden müssen.	Wenn das vorhandene Betriebssystem die gesamte Festplatte belegt, müssen Sie das vorhandene Betriebssystem so beibehalten, dass es neben der Solaris 10 10/08-Software auf dem System existieren kann. Von dieser Entscheidung hängt es ab, wie Sie das Schlüsselwort `fdisk`(1M) im Systemprofil angeben.	„x86: `fdisk`-Profilschlüsselwort (UFS und ZFS)“ auf Seite 134
Überprüfen Sie, ob das System unterstützt wird.	Informationen zur Systemunterstützung in der Solaris-Umgebung finden Sie in der Hardwaredokumentation.	Dokumentation des Hardwareherstellers
Überprüfen Sie, ob das System über genügend Festplattenspeicher für die Solaris-Software verfügt.	Überprüfen Sie, ob auf dem System ausreichend Festplattenspeicher für die Installation der Solaris-Software vorhanden ist.	Kapitel 4, „Systemvoraussetzungen, Richtlinien und Upgrades (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*
(Optional) Legen Sie die Systemparameter fest.	Sie können die Systeminformationen vorkonfigurieren und so vermeiden, dass Sie während des Installations- bzw. Upgrade-Vorgangs dazu aufgefordert werden, diese Informationen einzugeben.	Kapitel 2, „Vorkonfigurieren der Systemkonfigurationsinformationen (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
Bereiten Sie das System auf die benutzerdefinierte JumpStart-Installation vor.	Erstellen und validieren Sie eine `rules`-Datei und Profildateien.	Kapitel 3, „Vorbereiten von benutzerdefinierten JumpStart-Installationen (Vorgehen)“

TABELLE 6–3 x86: Übersicht der Schritte: Einrichten eines Systems für eine benutzerdefinierte JumpStart-Installation *(Fortsetzung)*

Aufgabe	Beschreibung	Siehe
(Optional) Bereiten Sie optionale benutzerdefinierte JumpStart-Funktionen vor.	Wenn Sie begin-Skripten, finish-Skripten oder andere optionale Funktionen nutzen wollen, bereiten Sie die Skripten bzw. Dateien vor.	Kapitel 4, „Verwenden der optionalen Funktionen der benutzerdefinierten JumpStart-Installation (Vorgehen)“ und Kapitel 5, „Erstellen von benutzerdefinierten Rule- und Probe-Schlüsselwörtern (Vorgehen)“
(Optional) Bereiten Sie die Installation der Solaris-Software über das Netzwerk vor.	Wenn Sie ein System von einem Remote-Abbild der Solaris Operating System for x86 Platforms-DVD oder der Solaris Software For x86 Platforms -CD installieren möchten, müssen Sie es so einrichten, dass es von einem Installations- oder einem Boot-Server aus gebootet und installiert werden kann.	Kapitel 6, „Installieren über das Netzwerk mithilfe von CDs (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*
(Optional) Bereiten Sie die Solaris Flash-Archivinstallation vor.	Vor der Installation eines Solaris Flash-Archivs sind bestimmte Schritte durchzuführen.	„So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor“ auf Seite 87
Führen Sie die Installation oder das Upgrade aus.	Booten Sie das System, um die Installation bzw. das Upgrade zu initiieren.	„x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus“ auf Seite 94

x86: Ausführen einer benutzerdefinierten JumpStart-Installation

Während einer benutzerdefinierten JumpStart-Installation versucht das JumpStart-Programm, das zu installierende System den Regeln in der Datei `rules.ok` zuzuordnen. Das JumpStart-Programm liest die Regeln nacheinander. Eine Übereinstimmung liegt vor, wenn das zu installierende System alle in der Regel definierten Systemattribute aufweist. Sobald ein System gefunden wird, das einer Regel entspricht, liest das JumpStart-Programm die Datei `rules.ok` nicht weiter und beginnt mit der Installation des Systems auf der Grundlage des in der Regel angegebenen Profils.

Sie können ein Solaris Flash-Archiv mit der benutzerdefinierten JumpStart-Installation installieren. Nähere Informationen hierzu finden Sie in „So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor“ auf Seite 87.

Wählen Sie eines der folgenden Verfahren:

- Informationen zum JumpStart-Standardverfahren finden Sie in „x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus“ auf Seite 94.
- Wie Sie eine benutzerdefinierte JumpStart-Installation durch Bearbeiten des GRUB-Befehls ausführen, erfahren Sie unter „x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB“ auf Seite 96.

▼ x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus

Verwenden Sie das folgende Verfahren, um das Betriebssystem Solaris für ein x86-basiertes System mit dem GRUB-Menü zu installieren.

1 Wenn das System Teil eines Netzwerks ist, stellen Sie sicher, dass ein Ethernet-Anschluss oder ein ähnlicher Netzwerkadapter mit dem System verbunden ist.

2 Wenn Sie ein System über eine `tip(1)`-Verbindung installieren möchten, muss das Fenster mindestens 80 Spalten breit und 24 Zeilen hoch sein.

Die aktuelle Größe des `tip`-Fensters können Sie mit dem Befehl `stty(1)` ermitteln.

3 Entscheiden Sie, ob Sie eine Profildiskette verwenden möchten.

Sie benötigen keine Profildiskette mehr zum Booten des Systems, es kann aber eine Diskette erstellt werden, die ausschließlich das JumpStart-Verzeichnis enthält. Diese können Sie dann zum Beispiel nutzen, um eine JumpStart-Installation auszuführen, wenn von CD-ROM gebootet wird.

- Wenn Sie eine Profildiskette verwenden, legen Sie diese in das Diskettenlaufwerk des Systems ein.
- Wenn Sie keine Profildiskette verwenden, fahren Sie mit Schritt 4 fort.

4 Entscheiden Sie, wie das System gebootet werden soll.

- Zum Booten von der Solaris Operating System-DVD oder Solaris Software-1-CD legen Sie nun den Datenträger ein. Das BIOS des Systems muss das Booten von einer DVD oder CD unterstützen.

- Wenn Sie über das Netzwerk booten, verwenden Sie PXE (Preboot Execution Environment). Das System muss PXE unterstützen. Aktivieren Sie die Unterstützung für PXE mit dem BIOS-Setup des Systems oder dem Konfigurationstool des Netzwerkadapters.

5 (Optional) Wenn Sie von einer DVD oder CD booten möchten, müssen Sie ggf. die Boot-Einstellung im System-BIOS entsprechend ändern. Erläuterungen dazu finden Sie in der Hardware-Dokumentation.

6 Wenn das System ausgeschaltet ist, schalten Sie es ein. Wenn das System eingeschaltet ist, starten Sie es neu.

Das GRUB-Menü wird angezeigt. Es zeigt eine Liste der Boot-Einträge an.

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------+
|Solaris 10 10/08 Abbildverzeichnis                                  |
|Solaris Serial Console ttya                                         |
|Solaris Serial Console ttyb (for lx50, v60x and v65x               |
+-------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

Abbildverzeichnis ist dabei der Name des Verzeichnisses, in dem das Installationsabbild gespeichert ist. Der Pfad zu den JumpStart-Dateien wurde mit dem Befehl `add_install_client` und der Option `-c` definiert.

Hinweis – Anstatt nun direkt über den GRUB-Eintrag zu booten, können Sie diesen bearbeiten. Nachdem Sie den GRUB-Eintrag bearbeitet haben, führen Sie dann die JumpStart-Installation aus. Wie Sie den GRUB-Eintrag bearbeiten und welche Installationsoptionen zur Verfügung stehen, erfahren Sie unter „x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB“ auf Seite 96.

---

7 Führen Sie an der Eingabeaufforderung einen der folgenden Schritte aus:

```
Select the type of installation you want to perform:

         1 Solaris Interactive
         2 Custom JumpStart
         3 Solaris Interactive Text (Desktop session)
         4 Solaris Interactive Text (Console session)
         5. Apply driver updates
         6. Single User Shell
Enter the number of your choice.
Please make a selection (1-6).
```

Um die benutzerdefinierte JumpStart-Installation auszuwählen, geben Sie **2** ein und drücken die Eingabetaste.

Anschließend beginnt die JumpStart-Installation.

Hinweis –

- Wenn Sie nicht innerhalb von 30 Sekunden eine Wahl treffen, startet das interaktive Solaris-Installationsprogramm. Sie können den Countdown stoppen, indem Sie in der Befehlszeile eine beliebige Taste drücken.
- Bei Auswahl der Optionen 1, 3 oder 4 erfolgt eine interaktive Installation. Informationen zur interaktiven Installation finden Sie in *Solaris 10 10/08 Installationshandbuch: Grundinstallationen*.
- Durch Auswahl der Option 5 installieren Sie Treiber-Updates.
- Über Option 6 können Sie verschiedene Wartungsaufgaben ausführen.

---

8 **Wenn Sie die Systeminformationen nicht in der Datei `sysidcfg` vorkonfiguriert haben, beantworten Sie die Fragen zur Systemkonfiguration.**

9 **Befolgen Sie die Anweisungen auf dem Bildschirm und installieren Sie die Software.**

Nach Abschluss der Solaris-Software durch das JumpStart-Programm wird das System automatisch neu gestartet. Außerdem wird die GRUB-Datei `menu.lst` automatisch aktualisiert. Beim nächsten Aufruf des GRUB-Menüs erscheint dann die von Ihnen installierte Solaris-Instanz.

Nach Abschluss der Installation werden die Installationsprotokolle in einer Datei gespeichert. Die Installationsprotokolle finden Sie in den folgenden Verzeichnissen:

- `/var/sadm/system/logs`
- `/var/sadm/install/logs`

x86: Ausführen einer benutzerdefinierten JumpStart-Installation durch Bearbeiten des Boot-Befehls in GRUB

In bestimmten Fällen, z. B. zur Fehlersuche, möchten Sie den GRUB-Boot-Befehl modifizieren. Das folgende Verfahren beschreibt, wie Sie den boot-Befehl in GRUB vor dem Ausführen einer benutzerdefinierten JumpStart-Installation bearbeiten.

▼ x86: So bearbeiten Sie den Boot-Befehl in GRUB

1 **Um die Installation zu starten, führen Sie Schritt 1 bis Schritt 5 des vorhergehenden Verfahrens aus („x86: So führen Sie eine Installation oder ein Upgrade mit dem benutzerdefinierten JumpStart-Programm und GRUB aus“ auf Seite 94).**

2 Wenn das System ausgeschaltet ist, schalten Sie es ein. Wenn das System eingeschaltet ist, starten Sie es neu.

Das GRUB-Menü wird angezeigt. Es zeigt eine Liste der Boot-Einträge an. Bei dem angebotenen Eintrag handelt es sich um die Solaris-Instanz, die installiert werden soll.

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-------------------------------------------------------------------+
|Solaris 10 10/08 Abbildverzeichnis                                 |
|Solaris Serial Console ttya                                        |
|Solaris Serial Console ttyb (lx50, v60x and v68)                   |
+-------------------------------------------------------------------+
Use the ^ and v keys to select which entry is highlighted. Press
enter to boot the selected OS, 'e' to edit the commands before
booting, or 'c' for a command-line.
```

Abbildverzeichnis ist dabei der Name des Verzeichnisses, in dem das Installationsabbild gespeichert ist.

Hinweis –

- Wenn Sie den Pfad zum JumpStart-Verzeichnis per NFS über den Befehl `add_install_client` mit der Option `-c` angegeben haben, müssen Sie den Pfad nicht in den Boot-Eintrag aufnehmen.
- Wenn Sie hingegen kein NFS verwenden, müssen Sie den Pfad zu der komprimierten Konfigurationsdatei angeben, die das JumpStart-Verzeichnis enthält.

---

3 Um den Bootvorgang zu unterbrechen und den Menüeintrag zu bearbeiten, geben Sie e ein.

Das GRUB-Bearbeitungsmenü wird angezeigt.

```
kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B console=ttyb,\
install_media=131.141.2.32:/export/mary/v11 \
module /I86PC.Solaris_11-8/x86.new
```

4 Wählen Sie den Boot-Eintrag mithilfe der Pfeiltasten aus.

5 Um den ausgewählten Befehl zu bearbeiten, geben Sie e ein.

Ein der folgenden Ausgabe ähnlicher Befehl wird angezeigt.

```
grub edit>kernel /I86PC.Solaris_11-8/multiboot kernel/unix -B \
console=ttyb,install_media=131.141.2.32:/export/mary/_\
module /I86PC.Solaris_11-8/x86.new
```

6 Passen Sie den Befehl je nach Bedarf mit den nötigen Optionen an.

Die Syntax für eine JumpStart-Installation lautet wie folgt:

```
grub edit>kernel /I86PC.Solaris_11-Abbildverzeichnis/multiboot kernel/unix/ \
- install [URL|ask] options -B install_media=Medientyp
```

Eine Beschreibung der JumpStart-Optionen finden Sie unter „x86: Befehlsreferenz zum Booten des Systems“ auf Seite 98.

Im folgenden Beispiel wird das Betriebssystem mit einem benutzerdefinierten JumpStart-Profil über das Netzwerk installiert.

```
kernel /I86PC.Solaris_11-8/multiboot kernel/unix/ - install \
-B install_media=131.141.2.32:/export/mary/v11 \
module /I86PC.Solaris_11-8/x86.new
```

7 Um die Änderungen zu übernehmen, drücken Sie die Eingabetaste.

Ihre Änderungen werden gespeichert, und das GRUB-Hauptmenü erscheint erneut.

Hinweis – Um zum GRUB-Hauptmenü zurückzukehren, ohne Ihre Änderungen zu speichern, drücken Sie stattdessen die Esc-Taste.

---

8 Um die Installation zu starten, geben Sie b ein.

x86: Befehlsreferenz zum Booten des Systems

In der folgenden Tabelle sind die Befehlszeilenoptionen für den Boot-Befehl im GRUB-Menü beschrieben. Die aufgeführten Optionen eignen sich für eine JumpStart-Installation.

Die Syntax des Boot-Befehls lautet wie folgt:

```
kernel /I86PC.Solaris_11-Abbildverzeichnis/multiboot kernel/unix/ - install \
[URL|ask] options -B install_media=Medientyp
```

TABELLE 6–4 Befehlsreferenz zum Boot-Befehl im GRUB-Menü

Option	Beschreibung

`- install`	Führt eine benutzerdefinierte JumpStart-Installation aus.
	
	Im folgenden Beispiel bootet das System von DVD, wobei die folgenden Optionen verwendet werden:
	■ `install` startet eine benutzerdefinierte JumpStart-Installation.
	■ `file://jumpstart/config.tar` verweist auf das JumpStart-Profil auf dem lokalen Datenträger.
	
	`kernel /I86pc.Solaris_11.8/multiboot - install file://jumpstart/config.tar \`
	`-B install_media=dvdrom module /I86Solaris_11.8/x86.new`

TABELLE 6–4 Befehlsreferenz zum Boot-Befehl im GRUB-Menü *(Fortsetzung)*

Option	Beschreibung

[*URL*| ask]	Gibt die Adresse der benutzerdefinierten JumpStart-Dateien an oder fordert zu deren Eingabe auf.
	■ *URL* – Gibt den Pfad zu den Dateien an. Sie können einen URL für Dateien angeben, die auf einem HTTP- oder HTTPS-Server abgelegt sind.
	Die Syntax bei einem HTTP-Server lautet:
	`http://`*Servername:IP-Adresse/Jumpstart_Verz-Pfad/ Komprimierte_Konfigurationsdatei&Proxy_Info*
	■ Wenn Sie eine `sysidcfg`-Datei in die komprimierte Konfigurationsdatei aufgenommen haben, müssen Sie wie im folgenden Beispiel die IP-Adresse des Servers angeben, auf dem sich die Datei befindet:
	`kernel /I86pc.Solaris_11.8/multiboot` **`install \`**
	`http://192.168.2.1/jumpstart/config.tar \`
	`-B install_media=192.168.2.1/export/Solaris_11.8/boot \`
	`module /I86PC.Solaris_11.8/x86.new`
	■ Wenn Sie die komprimierte Konfigurationsdatei auf einem HTTP-Server hinter einer Firewall gespeichert haben, müssen Sie während des Boot-Vorgangs einen Proxy-Server angeben. Sie brauchen keine IP-Adresse für den Server anzugeben, auf dem sich die Datei befindet. Sie müssen jedoch wie im folgenden Beispiel eine IP-Adresse für den Proxy-Server angeben:
	`kernel /I86pc.Solaris_11.8/multiboot` **`install \`**
	`http://www.shadow.com/jumpstart/config.tar&proxy=131.141.6.151 \`
	`-B install_media=192.168.2.1/export/Solaris_11.8/boot \`
	`module /I86PC.Solaris_11.8/x86.new`
	■ `ask` – Legt fest, dass Sie das Installationsprogramm zur Eingabe der Adresse der komprimierten Konfigurationsdatei auffordert. Die Aufforderung erscheint, nachdem das System gestartet und die Verbindung zum Netzwerk hergestellt wurde. Bei Verwendung dieser Option können Sie keine vollständig automatische JumpStart-Installation durchführen.
	Wenn Sie durch Drücken der Eingabetaste die Eingabeaufforderung umgehen, konfiguriert das Solaris-Installationsprogramm die Netzwerkparameter interaktiv. Danach fordert Sie das Installationsprogramm zur Eingabe des Speicherorts der komprimierten Konfigurationsdatei auf.
	Im folgenden Beispiel wird eine benutzerdefinierte JumpStart-Installation ausgeführt und von DVD gebootet. Sobald das System eine Verbindung zum Netzwerk aufgebaut hat, werden Sie gebeten, den Speicherort der Konfigurationsdatei anzugeben.
	`kernal /boot/multiboot kernel/unix` **`install ask`** `-B \`
	`install_media=192.168.2.1:export/sol_11_x86/boot module \`
	`/I86PC.Solaris_11.8_`

TABELLE 6–4 Befehlsreferenz zum Boot-Befehl im GRUB-Menü *(Fortsetzung)*

Option	Beschreibung

Optionen	■ `dhcp` – Legt fest, dass die zum Booten des Systems erforderlichen Netzwerkinformationen über einen DHCP-Server abgerufen werden. Diese Option wird bei einer JumpStart-Installation nicht benötigt. Wenn Sie `dhcp` weglassen und somit angeben, dass kein DHCP-Server verwendet werden soll, verwendet das System die Datei `/etc/bootparams` oder die Datenbank `bootparams` des Naming Service. Sie würden zum Beispiel nicht `dhcp` angeben, wenn Sie eine statische IP-Adresse beibehalten wollen. Beispiel:
	
	`kernel /I86pc.Solaris_11.8/multiboot` **`install`** `\`
	`dhcp` `-B install_media=192.168.2.1:/export/Solaris_11.8/ \`
	`boot module /I86PC.Solaris_11.8/x86.new`
	
	■ Die Optionen `nowin` und `text` sind für JumpStart-Installationen irrelevant. Sie sind bei interaktiven Installationen bedeutsam. Weitere Informationen finden Sie unter „So führen Sie eine Installation bzw. ein Upgrade mit dem Solaris-Installationsprogramm mit GRUB aus“ in *Solaris 10 10/08 Installationshandbuch: Grundinstallationen*.

◆ ◆ ◆ KAPITEL 7

Benutzerdefinierte JumpStart-Installation (Beispiele)

Dieses Kapitel bietet ein Beispiel für die Einrichtung und die Installation der Solaris-Software auf SPARC- und x86-basierten Systemen mithilfe des benutzerdefinierten JumpStart-Installationsverfahrens.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele.

- „Konfiguration des Beispielstandorts“ auf Seite 102
- „Erstellen eines Installationsservers“ auf Seite 103
- „x86: Erstellen eines Boot-Servers für die Marketingsysteme“ auf Seite 104
- „Erstellen eines JumpStart-Verzeichnisses“ auf Seite 105
- „Freigeben des JumpStart-Verzeichnisses zur gemeinsamen Nutzung“ auf Seite 105
- „SPARC: Erstellen des Profils für die Entwicklungsgruppe“ auf Seite 106
- „x86: Erstellen des Profils für die Marketinggruppe“ auf Seite 106
- „Aktualisieren der Datei `rules`“ auf Seite 107
- „Validieren der Datei `rules`“ auf Seite 108
- „SPARC: Einrichten der Entwicklungssysteme für die Installation über das Netzwerk“ auf Seite 108
- „x86: Einrichten der Marketingsysteme für die Installation über das Netzwerk“ auf Seite 109
- „SPARC: Booten der Entwicklungssysteme und Installation der Solaris-Software“ auf Seite 110
- „x86: Booten der Marketingsysteme und Installation der Solaris-Software“ auf Seite 110

Konfiguration des Beispielstandorts

Abbildung 7–1 zeigt die Standortkonfiguration für dieses Beispiel.

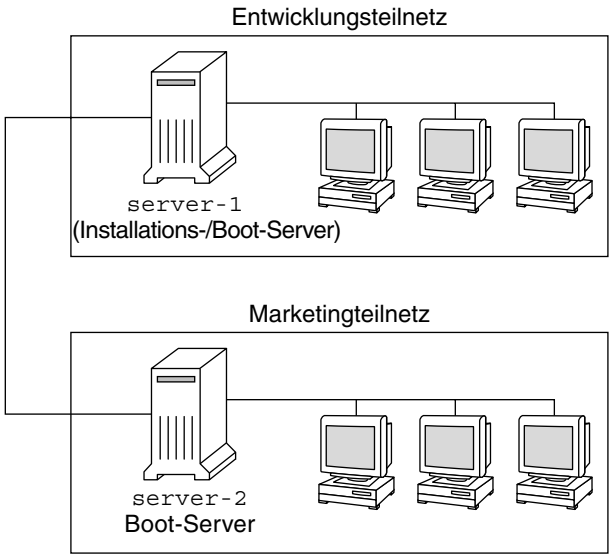

ABBILDUNG 7–1 Konfiguration des Beispielstandorts

An diesem Beispielstandort gelten folgende Bedingungen:

- SPARC: Die Engineering-Gruppe befindet sich in einem eigenen Teilnetz. Diese Gruppe setzt SPARCstation™-Systeme für die Softwareentwicklung ein.
- x86: Die Marketinggruppe befindet sich in einem eigenen Teilnetz. Diese Gruppe setzt x86-basierte Systeme für Textverarbeitung, Tabellenkalkulation und andere Office-Tools ein.
- Am Standort wird NIS genutzt. Die Ethernet-Adressen, IP-Adressen und Host-Namen der Systeme sind in den NIS-Maps vorkonfiguriert. Die Teilnetzmaske, Datum und Uhrzeit sowie die geografische Region für den Standort sind ebenfalls in den NIS-Maps vorkonfiguriert.

  Hinweis – Die Peripheriegeräte für die Marketingsysteme sind in der Datei `sysidcfg` vorkonfiguriert.

- Auf den Entwicklungs- und Marketingsystemen soll die 10 10/08-Software über das Netzwerk installiert werden.

Erstellen eines Installationsservers

Da bei beiden Gruppen die Installation der Solaris 10 10/08-Software über das Netzwerk erfolgen soll, erstellen Sie für beide Gruppen einen Installationsserver, `server-1`. Dazu kopieren Sie die Abbilder mit dem Befehl `setup_install_server(1M)` auf die lokale Festplatte von `server-1` (Verzeichnis `/export/install`). Sie können die Abbilder von unterschiedlichen Datenträgern kopieren:

- Solaris Software-CDs und Solaris Languages-CDs
- Solaris Operating System-DVD

Sie müssen das Abbild vom Datenträger in ein leeres Verzeichnis kopieren; in diesen Beispielen nach `sparc_10` und `x86_10`.

BEISPIEL 7–1 SPARC: Kopieren der 10 10/08-CDs

Legen Sie die Solaris Software for SPARC Platforms - 1-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# mkdir -p /export/install/sparc_10
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./setup_install_server /export/install/sparc_10
```

Legen Sie die Solaris Software for SPARC Platforms - 2-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/sparc_10
```

Wiederholen Sie diesen Befehl für jede Solaris Software, die Sie installieren möchten.

Legen Sie die SPARC: Solaris Languages for SPARC Platforms-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/sparc_10
```

Wiederholen Sie den vorigen Befehl für jede SPARC: Solaris Languages for SPARC Platforms-CD.

BEISPIEL 7–2 x86: Kopieren der 10 10/08-CDs

Legen Sie die Solaris Software for x86 Platforms - 1-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# mkdir -p /export/install/x86_10
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./setup_install_server /export/install/x86_10
```

BEISPIEL 7–2 x86: Kopieren der 10 10/08-CDs *(Fortsetzung)*

Legen Sie die Solaris Software for x86 Platforms - 2-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/x86_10
```

Wiederholen Sie diesen Befehl für jede Solaris Software, die Sie installieren möchten.

Legen Sie die erste Solaris Languages for x86 Platforms-CD in das CD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# cd /CD-Einhängepunkt/Solaris_10/Tools
server-1# ./add_to_install_server /export/install/x86_10
```

Wiederholen Sie den vorigen Befehl für jede Solaris Languages for x86 Platforms-CD.

BEISPIEL 7–3 SPARC: Kopieren der Solaris 10 10/08-DVD

Legen Sie die Solaris Operating System for SPARC Platforms-DVD in das DVD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# mkdir -p /export/install/sparc_10
server-1# cd /DVD-Einhängepunkt/Solaris_10/Tools
server-1# ./setup_install_server /export/install/sparc_10
```

BEISPIEL 7–4 x86: Kopieren der Solaris Operating System for x86 Platforms-DVD

Legen Sie die Solaris Operating System for x86 Platforms-DVD in das DVD-ROM-Laufwerk ein, das an `server-1` angeschlossen ist, und geben Sie die folgenden Befehle ein:

```
server-1# mkdir -p /export/install/x86_10
server-1# cd /DVD-Einhängepunkt/Solaris_10/Tools
server-1# ./setup_install_server /export/install/x86_10
```

x86: Erstellen eines Boot-Servers für die Marketingsysteme

Systeme können nicht von einem Installationsserver in einem anderen Teilnetz aus gebootet werden. Deshalb erstellen Sie im Teilnetz der Marketinggruppe einen Boot-Server, `server-2`. Dazu kopieren Sie mit dem Befehl `setup_install_server(1M)` die Boot-Software von der Solaris Operating System for x86 Platforms-DVD bzw. der Solaris Software for x86 Platforms - 1-CD. auf die lokale Festplatte von `server-2` (Verzeichnis `/export/boot`).

Zunächst wählen Sie also den Datenträger aus, von dem aus Sie die Boot-Software auf der lokalen Festplatte installieren möchten.

- Wenn Sie die Solaris Software for x86 Platforms - 1-CD in das CD-ROM-Laufwerk einlegen, das an `server-2` angeschlossen ist, geben Sie den folgenden Befehl ein:

  ```
  server-2# cd /CD-Einhängepunkt/Solaris_10/Tools
  server-2# ./setup_install_server -b /export/boot
  ```

- Wenn Sie die Solaris Operating System for x86 Platforms-DVD in das DVD-ROM-Laufwerk einlegen, das an `server-2` angeschlossen ist, geben Sie den folgenden Befehl ein:

  ```
  server-2# cd /DVD-Einhängepunkt/Solaris_10/Tools
  server-2# ./setup_install_server -b /export/boot
  ```

Die Option `b` des Befehls `setup_install_server` gibt an, dass der Befehl `setup_install_server` die Boot-Informationen in das Verzeichnis mit der Bezeichnung `/export/boot` kopieren soll.

Erstellen eines JumpStart-Verzeichnisses

Nachdem Sie den Installations- und den Boot-Server eingerichtet haben, erstellen Sie ein JumpStart-Verzeichnis auf `server-1`. Sie können dazu jedes System im Netzwerk verwenden. Dieses Verzeichnis enthält Dateien, die für eine benutzerdefinierte JumpStart-Installation der Solaris-Software erforderlich sind. Zum Einrichten dieses Verzeichnisses kopieren Sie das Beispielverzeichnis aus dem Abbild der Solaris Operating System-DVD oder dem Abbild der Solaris Software-1-CD, das Sie zuvor in `/export/install` kopiert haben:

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_10/Solaris_10/Misc/jumpstart_sample /jumpstart
```

Freigeben des JumpStart-Verzeichnisses zur gemeinsamen Nutzung

Damit die Systeme im Netzwerk Zugriff auf die Datei `rules` und die Profile erhalten, geben Sie das Verzeichnis `/jumpstart` zur gemeinsamen Nutzung frei. Damit ein Verzeichnis gemeinsam genutzt werden kann, müssen Sie die folgende Zeile in die Datei `/etc/dfs/dfstab` einfügen:

```
share -F nfs -o ro,anon=0 /jumpstart
```

Dann geben Sie in die Befehlszeile den Befehl `shareall` ein:

```
server-1# shareall
```

SPARC: Erstellen des Profils für die Entwicklungsgruppe

Für die Entwicklungssysteme erstellen Sie eine Datei mit dem Namen `eng_prof` im Verzeichnis `/jumpstart`. Die Datei `eng_prof` enthält die folgenden Einträge, die definieren, wie die 10 10/08-Software auf den Systemen der Entwicklungsgruppe installiert werden soll:

```
install_type  initial_install
system_type   standalone
partitioning  default
cluster       SUNWCprog
filesys       any 512 swap
```

Dieses Beispiel enthält die folgenden Installationsanweisungen:

`install_type`	Gibt an, dass es sich um eine Neuinstallation und nicht um ein Upgrade handelt.
`system_type`	Gibt an, dass es sich bei den Entwicklungssystemen um Standalone-Systeme handelt.
`partitioning`	Gibt an, dass die JumpStart-Software zur Installation der Solaris-Software auf den Entwicklungssystemen mit der Standardfestplattenpartitionierung arbeitet.
`cluster`	Gibt an, dass die Solaris-Softwaregruppe für Entwickler installiert werden soll.
`filesys`	Gibt an, dass alle Systeme in der Entwicklungsgruppe einen Swap-Bereich von 512 MB aufweisen sollen.

x86: Erstellen des Profils für die Marketinggruppe

Für die Marketing-Systeme erstellen Sie eine Datei mit dem Namen `marketing_prof` im Verzeichnis `/jumpstart`. Die Datei `marketing_prof` enthält die folgenden Einträge, die definieren, wie die Solaris 10 10/08-Software auf den Systemen der Marketinggruppe installiert werden soll:

```
install_type  initial_install
system_type   standalone
partitioning  default
cluster       SUNWCuser
package       SUNWaudio
```

Dieses Beispiel enthält die folgenden Installationsanweisungen:

`install_type`	Gibt an, dass es sich um eine Neuinstallation und nicht um ein Upgrade handelt.
`system_type`	Gibt an, dass es sich bei den Marketingsystemen um Standalone-Systeme handelt.
`partitioning`	Gibt an, dass die JumpStart-Software zur Installation der Solaris-Software auf den Marketingsystemen mit der Standardfestplattenpartitionierung arbeiten soll.
`cluster`	Gibt an, dass die Solaris-Softwaregruppe für Endanwender installiert werden soll.
`package`	Gibt an, dass das Package für die Audio-Demo-Software auf allen Systemen installiert werden soll.

Aktualisieren der Datei `rules`

Jetzt müssen Sie Regeln in die Datei `rules` einfügen. Das Solaris-Installationsprogramm wählt anhand der Regeln während der benutzerdefinierten JumpStart-Installation das richtige Installationsprofil für die einzelnen Systeme aus.

Bei diesem Standort befinden sich die beiden Abteilungen in einem eigenen *Teilnetz* und weisen eigene Netzwerkadressen auf. Die Entwicklerabteilung befindet sich im Teilnetz 255.222.43.0, die Marketing-Abteilung in 255.222.44.0. Anhand dieser Angaben können Sie die Installation der Solaris 10 10/08-Software auf den Entwicklungs- und Marketing-Systemen steuern. Dazu bearbeiten Sie im Verzeichnis `/jumpstart` die Datei `rules`, löschen alle Beispielregeln und fügen die folgenden Zeilen in die Datei ein:

```
network 255.222.43.0 - eng_prof -
network 255.222.44.0 - marketing_prof -
```

Diese Regeln legen fest, dass die Installation der 10 10/08-Software auf den Systemen im Netzwerk 255.222.43.0 unter Verwendung des Profils `eng_prof` erfolgen soll. Auf den Systemen im Netzwerk 255.222.44.0 soll die Installation der 10 10/08-Software dagegen unter Verwendung des Profils `marketing_prof` erfolgen.

Hinweis – Sie können die Beispielregeln verwenden und zur Identifizierung der Systeme, auf denen die 10 10/08-Software unter Verwendung der Profile `eng_prof` und `marketing_prof` installiert werden soll, eine Netzwerkadresse angeben. Außerdem können Sie Host-Namen, Speichergröße und Modelltyp als Rule-Schlüsselwort verwenden. Tabelle 8–1 enthält eine vollständige Liste der Schlüsselwörter, die in einer `Rules`-Datei verwendet werden können.

Validieren der Datei rules

Nachdem Sie die Datei rules und die Profile eingerichtet haben, führen Sie das Skript check aus, um die Gültigkeit der Dateien zu verifizieren:

```
server-1# cd /jumpstart
server-1# ./check
```

Wenn das Skript check keine Fehler findet, erstellt es die Datei rules.ok.

SPARC: Einrichten der Entwicklungssysteme für die Installation über das Netzwerk

Nachdem Sie das Verzeichnis /jumpstart und die erforderlichen Dateien eingerichtet haben, richten Sie die Entwicklungssysteme mit dem Befehl add_install_client auf dem Installationsserver (server-1) so ein, dass die Solaris-Software vom Installationsserver aus installiert wird. server-1 ist außerdem der Boot-Server für das Teilnetz der Entwicklungsgruppe.

```
server-1# cd /export/install/sparc_10/Solaris_10/Tools
server-1# ./add_install_client -c server-1:/jumpstart host-eng1 sun4u
server-1# ./add_install_client -c server-1:/jumpstart host-eng2 sun4u
```

Die im Befehl add_install_client verwendeten Optionen haben folgende Bedeutung:

-c — Gibt den Server (server-1) und den Pfad (/jumpstart) für das JumpStart-Verzeichnis an. Verwenden Sie diese Option, wenn Sie mit NFS arbeiten.

Hinweis – Wenn Sie nicht mit NFS arbeiten, geben Sie den Pfad zum JumpStart-Verzeichnis mit den folgenden Befehlen an:

- **Für SPARC-basierte Systeme** geben Sie den Pfad zusammen mit dem Befehl boot an.
- **Für x86-basierte Systeme** bearbeiten Sie zur Angabe des Pfads den Eintrag im GRUB-Menü

---

host-eng1 — Der Namen eines Systems in der Entwicklungsgruppe.

host-eng2 — Der Namen eines weiteren Systems in der Entwicklungsgruppe.

sun4u — Gibt die Plattformgruppe der Systeme an, die server-1 als Installationsserver nutzen. Die Plattformgruppe gilt für Ultra 5-Systeme.

x86: Einrichten der Marketingsysteme für die Installation über das Netzwerk

Als Nächstes führen Sie den Befehl `add_install_client` auf dem Boot-Server (`server-2`) aus. Dadurch werden die Marketingsysteme so eingerichtet, dass sie über den Boot-Server starten und die Solaris-Software vom Installationsserver (`server-1`) installieren:

```
server-2# cd /marketing/boot-dir/Solaris_10/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart host-mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart host-mkt2 i86pc
server-2# ./add_install_client -d -s server-1:/export/install/x86_10 \
-c server-1:/jumpstart SUNW.i86pc i86pc
server-2# ./add_install_client -c server-1:/jumpstart host-mkt1 sun4u
server-2# ./add_install_client -c server-1:/jumpstart host-mkt2 sun4u
```

Die im Befehl `add_install_client` verwendeten Optionen haben folgende Bedeutung:

-d
: Gibt an, dass der Client die Parameter für die Installation über das Netzwerk über DHCP abrufen soll. Diese Option ist für Clients erforderlich, die per PXE über das Netzwerk booten sollen. Für Clients, die zum Netzwerkstart nicht PXE verwenden, ist die Option `-d` nicht erforderlich.

-s
: Gibt den Installationsserver (`server-1`) und den Pfad zur Solaris-Software (`/export/install/x86_10`) an.

-c
: Gibt den Server (`server-1`) und den Pfad (`/jumpstart`) für das JumpStart-Verzeichnis an. Verwenden Sie diese Option, wenn Sie mit NFS arbeiten.

Hinweis – Wenn Sie nicht mit NFS arbeiten, geben Sie den Pfad zum JumpStart-Verzeichnis mit den folgenden Befehlen an:

- **Für SPARC-basierte Systeme** geben Sie den Pfad zusammen mit dem Befehl `boot` an.
- **Für x86-basierte Systeme** bearbeiten Sie zur Angabe des Pfads den Eintrag im GRUB-Menü

---

host-mkt1
: Der Namen eines Systems in der Marketinggruppe.

host-mkt2
: Der Namen eines weiteren Systems in der Marketinggruppe.

sun4u
: Gibt die Plattformgruppe der Systeme an, die `server-1` als Installationsserver nutzen. Die Plattformgruppe gilt für Ultra 5-Systeme.

SUNW.i86pc	Der DHCP-Klassenname für alle Solaris-x86-Clients. Verwenden Sie diesen Klassennamen, wenn Sie alle Solaris-x86-DHCP-Clients mit einem einzigen Befehl konfigurieren wollen.
`i86pc`	Gibt die Plattformgruppe der Systeme an, die diesen Boot-Server verwenden. Der Plattformname steht für x86-basierte Systeme.

SPARC: Booten der Entwicklungssysteme und Installation der Solaris-Software

Nachdem Sie die Server und Dateien eingerichtet haben, können Sie die Entwicklungssysteme mit dem folgenden `boot`-Befehl an der Eingabeaufforderung `ok` (PROM) der einzelnen Systeme booten:

```
ok boot net - install
```

Solaris wird automatisch auf den Systemen der Entwicklungsgruppe installiert.

x86: Booten der Marketingsysteme und Installation der Solaris-Software

Sie haben die folgenden Möglichkeiten, das System zu booten:

- Solaris Software for x86 Platforms - 1-CD
- Solaris Operating System for x86 Platforms-DVD
- Netzwerk (mithilfe von PXE)

Die Solaris-Software wird automatisch auf den Systemen der Marketinggruppe installiert.

◆ ◆ ◆ KAPITEL 8

Benutzerdefinierte JumpStart-Installation (Referenz)

In diesem Kapitel sind die Schlüsselwörter und Werte aufgeführt, die Sie in der Datei `rules`, Profilen sowie Begin- und Finish-Skripten verwenden können.

Hinweis – Wenn Sie ein Solaris-ZFS-Root-Pool installieren, finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“ Erläuterungen zu Einschränkungen sowie Profilbeispiele. Weiterhin finden Sie eine Liste ZFS-spezifischer Schlüsselwörter sowie Schlüsselwörter, die in einem Profil verwendet werden können, in Tabelle 8–2.

- „Rule-Schlüsselwörter und -Werte“ auf Seite 111
- „Profilschlüsselwörter und -werte“ auf Seite 116
- „Benutzerdefinierte JumpStart-Umgebungsvariablen“ auf Seite 166
- „Probe-Schlüsselwörter und -Werte“ auf Seite 168

Rule-Schlüsselwörter und -Werte

Tabelle 8–1 beschreibt die Schlüsselwörter und Werte, die in der `rules`-Datei verwendet werden können. Nähere Informationen zum Erstellen einer `rules`-Datei finden Sie unter „Erstellen der Datei `rules`“ auf Seite 34.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten

Schlüsselwort	Wert	Entspricht
`any`	Minuszeichen (`-`)	Allem. Das Schlüsselwort `any` ist immer wirksam.
`arch`	*Prozessortyp* Gültige Werte für *Prozessortyp*: ■ SPARC: `sparc` ■ x86: `i386`	Dem Prozessortyp eines Systems. Mit dem Befehl `uname -p` können Sie den Prozessortyp eines Systems abfragen.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten *(Fortsetzung)*

Schlüsselwort	Wert	Entspricht

`disksize`	*Name_der_Festplatte Größenbereich*
	
	Name_der_Festplatte – Ein Festplattenname im Format `c` *x*`t`*y*`d` *z*, zum Beispiel `c0t3d0` oder `c0d0`, oder das reservierte Wort `rootdisk`. Wenn Sie `rootdisk` verwenden, werden die Festplatten in der folgenden Reihenfolge nach der entsprechenden Festplatte durchsucht:
	■ SPARC: Die Festplatte mit dem vorinstallierten Boot-Abbild. Hierbei handelt es sich um ein neues SPARC-System mit ab Werk installiertem JumpStart.
	■ Die Festplatte `c0t3d0s0`, sofern vorhanden.
	■ Die erste verfügbare Festplatte, die in der Kernel-Sondierungsreihenfolge gesucht wird.
	
	Größenbereich – Die Größe der Festplatte, die als ein Bereich von MB angegeben werden muss (*x*- *x*).
	
	Hinweis – Bedenken Sie beim Berechnen des *Größenbereichs*, dass ein MB 1.048.576 Byte entspricht. Eine Festplatte, die laut Hersteller "535 MB" aufweist, hat möglicherweise nur eine Kapazität von 510 Millionen Bytes. Das JumpStart-Programm zeigt die 535 MB-Festplatte als 510 MB-Festplatte an, da 535.000.000 / 1.048.576 = 510 ist. Eine 535-MB-Festplatte entspricht nicht einem *Größenbereich* von 530–550.
		Dem Namen und der Größe einer Festplatte in MB.
		
		Beispiel:
		
		`disksize c0t3d0 250-300`
		
		In diesem Beispiel sucht das JumpStart-Programm eine Festplatte mit dem Namen `c0t3d0`. Diese Festplatte soll eine Kapazität von 250 bis 300 MB haben.
		
		Beispiel:
		
		`disksize rootdisk 750-1000`
		
		In diesem Beispiel sucht das JumpStart-Programm in der folgenden Reihenfolge nach einer passenden Festplatte:
		1. Eine Systemfestplatte, die ein vorinstalliertes Boot-Abbild enthält.
		2. Die Festplatte `c0t3d0s0`, sofern vorhanden.
		3. Die erste verfügbare Festplatte mit einer Kapazität von 750 MB bis 1 GB.

`domainname`	*Domänenname*	Ein Systemdomänenname, der festlegt, wie ein Namen-Service Informationen ermittelt. Bei einem bereits installierten System können Sie den Domänennamen mit dem Befehl `domainname` abrufen.
`hostaddress`	*IP-Adresse*	Der IP-Adresse eines Systems.
`hostname`	*Host-Name*	Der Host-Name eines Systems. Bei einem bereits installierten System können sie den Host-Namen mit dem Befehl `uname -n` abrufen.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten *(Fortsetzung)*

Schlüsselwort	Wert	Entspricht
`installed`	*Speicherbereich Version* *Speicherbereich* – Ein Name für einen Festplatten-Speicherbereich im Format c*w*t*x*d*y*s*z*, zum Beispiel `c0t3d0s5` oder eines der reservierten Wörter `any` oder `rootdisk`. Wenn Sie `any` verwenden, durchsucht das JumpStart-Programm alle Festplatten des Systems in der Kernel-Sondierungsreihenfolge. Wenn Sie `rootdisk` verwenden, werden die Festplatten in der folgenden Reihenfolge nach der entsprechenden Festplatte durchsucht: ■ SPARC: Die Festplatte mit dem vorinstallierten Boot-Abbild. Hierbei handelt es sich um ein neues SPARC-System mit ab Werk installiertem JumpStart. ■ Die Festplatte `c0t3d0s0`, sofern vorhanden. ■ Die erste verfügbare Festplatte, die in der Kernel-Sondierungsreihenfolge gesucht wird. *Version* – Eine Versionsbezeichnung oder eines der reservierten Wörter `any` und `upgrade`. Wenn Sie `any` verwenden, gilt jedes Solaris- oder SunOS-Release als Entsprechung. Wenn Sie `upgrade` verwenden, gilt jedes unterstützte Solaris-Release als Entsprechung, bei dem ein Upgrade möglich ist. Wenn das JumpStart-Programm ein Solaris-Release findet, aber die Version nicht ermitteln kann, wird die Version `SystemV` zurückgegeben.	Einer Festplatte mit einem Root-Dateisystem (`/`), das einer bestimmten Version der Solaris-Software entspricht. Beispiel: `installed c0t3d0s1 Solaris 10` In diesem Beispiel sucht das JumpStart-Programm ein System mit einem Solaris-Root-Dateisystem (`/`) auf `c0t3d0s1`.
`karch`	*Plattformgruppe* Gültige Werte: `sun4u`, `i86pc` und `prep`. Eine Liste der Systeme mit dazugehöriger Plattformgruppe ist im *Solaris Handbuch zur Hardware-Plattform von Sun* unter `http://docs.sun.com` enthalten.	Der Plattformgruppe eines Systems. Bei einem bereits installierten System können Sie mit dem Befehl `arch -k` oder `uname -m` die Plattformgruppe des Systems abrufen.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten *(Fortsetzung)*

Schlüsselwort	Wert	Entspricht
`memsize`	*physischer_Hauptspeicher* Sie müssen den Wert als einen Bereich von MB (*x*- *x*) oder in Form eines einzigen MB-Werts angeben.	Der physischen Hauptspeichergröße eines Systems in MB. Beispiel: `memsize 64-128` Bei diesem Beispiel wird ein System mit einer physischen Hauptspeichergröße von 64 bis 128 MB gesucht. Bei einem bereits installierten System können Sie mit dem Befehl `prtconf` die physische Hauptspeichergröße des Systems abrufen.
`model`	*Plattformname*	Der Plattformname eines Systems. Eine Liste der zulässigen Plattformnamen entnehmen Sie bitte dem *Solaris Handbuch zur Hardware-Plattform von Sun* unter·`http://docs.sun.com`. Bei einem bereits installierten System können Sie den Plattformnamen mit dem Befehl `uname -i` oder anhand der Ausgabe des Befehls `prtconf` (Zeile 5) ermitteln. **Hinweis –** Wenn der *Plattformname* Leerzeichen enthält, müssen Sie diese durch einen Unterstrich (_) ersetzen. Beispiel: SUNW,Sun_4_50
`network`	*Netzwerknummer*	Der Netzwerknummer eines System, die das JumpStart-Programm mithilfe eines logischen UND zwischen der IP-Adresse und der Teilnetzmaske des Systems ermittelt. Beispiel: `network 192.168.2.0` In diesem Beispiel wird ein System gesucht, das die IP-Adresse 192.168.2.8 aufweist, wenn die Teilnetzmaske 255.255.255.0 lautet.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten *(Fortsetzung)*

Schlüsselwort	Wert	Entspricht
`osname`	`Solaris_`*x*	Einer Version der Solaris-Software, die bereits auf einem System installiert ist. Beispiel: `osname Solaris 10` In diesem Beispiel sucht das JumpStart-Programm ein System, auf dem das Betriebssystem Solaris 10 10/08 bereits installiert ist.
`probe`	*Probe-Schlüsselwort*	Einem gültigen Probe-Schlüsselwort oder einem gültigen benutzerdefinierten Probe-Schlüsselwort. Beispiel: `probe disks` Dieses Beispiel gibt die Größe der Festplatten eines Systems in MB zurück, und zwar in der Kernel-Sondierungsreihenfolge, zum Beispiel `c0t3d0s1`, `c0t4d0s0` auf einem SPARC-System. Das JumpStart-Programm setzt die Umgebungsvariablen `SI_DISKLIST`, `SI_DISKSIZES`, `SI_NUMDISKS` und `SI_TOTALDISK`. **Hinweis –** Das Schlüsselwort `probe` stellt insofern eine Ausnahme dar, als nicht versucht wird, ein System mit entsprechenden Attributen zu finden und ein Profil auszuführen. Das Schlüsselwort `probe` gibt stattdessen einen Wert zurück. Deshalb können Sie beim Rule-Schlüsselwort `probe` keine Begin-Skripte, Profile oder Finish-Skripte angeben. Probe-Schlüsselwörter werden in Kapitel 5, „Erstellen von benutzerdefinierten Rule- und Probe-Schlüsselwörtern (Vorgehen)“ beschrieben.

TABELLE 8–1 Beschreibung von Rule-Schlüsselwörtern und -Werten *(Fortsetzung)*

Schlüsselwort	Wert	Entspricht
`totaldisk`	*Größenbereich* Sie müssen den Wert als einen Bereich von MB (*x- x*) angeben. **Hinweis –** Bedenken Sie beim Berechnen des *Größenbereichs*, dass ein MB 1.048.576 Byte entspricht. Eine Festplatte, die laut Hersteller "535 MB" aufweist, hat möglicherweise nur eine Kapazität von 510 Millionen Bytes. Das JumpStart-Programm zeigt die 535 MB-Festplatte als 510 MB-Festplatte an, da 535.000.000 / 1.048.576 = 510 ist. Eine 535-MB-Festplatte entspricht nicht einem *Größenbereich* von 530–550.	Dem gesamten Festplattenspeicher eines Systems in MB. Diese Angabe umfasst alle betriebsbereiten Festplatten, die an das System angeschlossen sind. Beispiel: `totaldisk 300-500` In diesem Beispiel sucht das JumpStart-Programm ein System mit einem Gesamtfestplattenspeicher von 300 bis 500 MB.

Profilschlüsselwörter und -werte

In diesem Abschnitt werden die Profilschlüsselwörter und -werte beschrieben, die Sie in einem Profil verwenden können. Nähere Informationen zum Erstellen eines Profils finden Sie unter „Erstellen eines Profils“ auf Seite 38. Diese Schlüsselwörter dienen zur Installation von UFS- und ZFS-Dateisystemen. Wenn ein Schlüsselwort in einem ZFS-Profil verwendet werden kann, ist es mit „ZFS“ gekennzeichnet.

Profilschlüsselwörter im Überblick

Tabelle 8–2 bietet eine schnelle Möglichkeit, zu bestimmen, welche Schlüsselwörter basierend auf Ihrer Installationsumgebung verwendet werden können. Sofern in der Schlüsselwortbeschreibung nicht anders angegeben, kann das Schlüsselwort nur für die Neuinstallationsoption eingesetzt werden. Diese Schlüsselwörter sind darüber hinaus nur für UFS-Dateisysteme bestimmt, falls nicht explizit vermerkt ist, dass das betreffende Schlüsselwort auch in ZFS-Root-Profilen verwendet werden kann.

TABELLE 8–2 Übersicht über Profilschlüsselwörter

	Installationsszenarien					
Profilschlüsselwort	Standalone-System (nicht vernetzt)	Standalone-System (vernetzt) oder Server	OS-Server	Aufrüstung	Upgrade mit Neuzuweisung von Festplattenspeicher	Kann für ein ZFS-Root-Pool verwendet werden
`archive_location` (Installation von Solaris Flash-Archiven)	X	X				

TABELLE 8–2 Übersicht über Profilschlüsselwörter *(Fortsetzung)*

	Installationsszenarien					
Profilschlüsselwort	**Standalone-System (nicht vernetzt)**	**Standalone-System (vernetzt) oder Server**	**OS-Server**	**Aufrüstung**	**Upgrade mit Neuzuweisung von Festplattenspeicher**	**Kann für ein ZFS-Root-Pool verwendet werden**
`backup_media`					X	
`boot_device` (UFS und ZFS)	X	X	X			X
`bootenv` (UFS und ZFS)	X	X	X			X
`client_arch`			X			
`client_root`			X			
`client_swap`			X			
`cluster` (Hinzufügen von Softwaregruppen) (UFS und ZFS)	X	X	X			X
`cluster` (Hinzufügen bzw. Löschen von Clustern) (UFS und ZFS)	X	X	X	X	X	X
`dontuse` (UFS und ZFS)	X	X	X			X
`fdisk` (nur x86) (UFS und ZFS)	X	X	X			X
`filesys` (Einhängen von Remote-Dateisystemen) (UFS und ZFS)		X	X			X
`filesys` (Erstellen lokaler Dateisysteme)	X	X	X			
`filesys` (Erstellen gespiegelter Dateisysteme)	X	X	X			
`forced_deployment` (Installation von Solaris Flash-Differenzarchiven)	X	X				
`geo` (UFS und ZFS)	X	X	X	X	X	X
`install_type` (UFS und ZFS)	X	X	X	X	X	X
`layout_constraint`					X	
`local_customization` (Installation von Solaris Flash-Archiven)	X	X				

TABELLE 8–2 Übersicht über Profilschlüsselwörter *(Fortsetzung)*

	Installationsszenarien					
Profilschlüsselwort	**Standalone-System (nicht vernetzt)**	**Standalone-System (vernetzt) oder Server**	**OS-Server**	**Aufrüstung**	**Upgrade mit Neuzuweisung von Festplattenspeicher**	**Kann für ein ZFS-Root-Pool verwendet werden**
`locale` (UFS und ZFS)	X	X	X	X	X	X
`metadb` (Erzeugen von Statusdatenbankreplikationen)	X	X	X			
`no_master_check` (Installation von Solaris Flash-Differenzarchiven)	X	X				
`no_content_check` (Installation von Solaris Flash-Differenzarchiven)	X	X				
`num_clients`			X			
`package` (UFS und ZFS)	X	X	X	X	X	X
`partitioning`	X	X	X			
`patch`	X	X	X	X	X	
`pool` (nur ZFS-Root-Pools)	X	X	X			X
`root_device` (UFS und ZFS)	X	X	X	X	X	X
`system_type`	X	X	X			
`usedisk` (UFS und ZFS)	X	X	X			X

Profilschlüsselwörter - Beschreibung und Beispiele

`archive_location`-Schlüsselwort

`archive_location` *Abrufmethode Speicherort*

Abrufmethode Die Werte von *Abrufmethode* und *Position* sind abhängig vom Speicherort des Solaris Flash-Archivs. In den folgenden Abschnitten werden die Werte, die Sie für *Abrufmethode* und *Position* angeben können, sowie Beispiele für die Verwendung des Schlüsselworts `archive_location` beschrieben.

- „NFS-Server“ auf Seite 119
- „HTTP- oder HTTPS-Server“ auf Seite 119
- „FTP-Server“ auf Seite 121
- „Lokales Bandlaufwerk“ auf Seite 122
- „Lokales Gerät“ auf Seite 123

- „Lokale Datei“ auf Seite 124

Position In den folgenden Abschnitten werden abhängig von der Position die relevanten Angaben beschrieben.

Achtung – Solaris Flash-Archive können auf Systemen mit installierten nicht-globalen Zonen nicht ordnungsgemäß erstellt werden. Die Solaris Flash-Funktion ist nicht mit der Solaris Zones-Partitionierungstechnologie kompatibel. Wenn Sie ein Solaris Flash-Archiv erstellen, wird dieses Archiv nicht korrekt installiert, wenn es unter den folgenden Bedingungen bereitgestellt wird:

- Das Archiv wird in einer nicht-globalen Zone erstellt.
- Das Archiv wird in einer globalen Zone erstellt, in der nicht-globale Zonen installiert sind.

NFS-Server

Ist das Archiv auf einem NFS-Server gespeichert, verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

`archive_location nfs` *Servername:/Pfad/Dateiname retry n*

Server Der Name des Servers, auf dem Sie das Archiv gespeichert haben.

Pfad Die Speicherposition des vom angegebenen Server abzurufenden Archivs. Enthält der Pfad $HOST, so ersetzen die Solaris Flash-Installationsdienstprogramme $HOST durch den Namen des Klonsystems, das gerade installiert wird.

Dateiname Der Name der Solaris Flash-Archivdatei.

retry n Ein optionales Schlüsselwort. *n* legt fest, wie oft die Solaris Flash-Dienstprogramme maximal versuchen, das Archiv einzuhängen.

BEISPIEL 8–1 NFS-Server

```
archive_location nfs golden:/archives/usrarchive

archive_location nfs://golden/archives/usrarchive
```

HTTP- oder HTTPS-Server

Ist das Archiv auf einem HTTP-Server gespeichert, verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

`archive_location http://`*Servername:Port/Pfad/Dateiname Optionale_Schlüsselworte*

Ist das Archiv auf einem HTTPS-Server gespeichert, verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

`archive_location https://`*Port/Pfad/Dateiname Optionale_Schlüsselworte*

Server	Der Name des Servers, auf dem Sie das Archiv gespeichert haben.
Port	Ein optionaler Port (Anschluss). *Port* kann eine Portnummer oder der Name eines TCP-Dienstes mit einer Portnummer sein, die bei der Ausführung ermittelt wird. Wenn Sie keinen Port angeben, verwenden die Solaris Flash-Installationsdienstprogramme den HTTP-Standardport, Nummer 80.
Pfad	Die Speicherposition des vom angegebenen Server abzurufenden Archivs. Enthält der Pfad $HOST, so ersetzen die Solaris Flash-Installationsdienstprogramme $HOST durch den Namen des Klonsystems, das gerade installiert wird.
Dateiname	Der Name der Solaris Flash-Archivdatei.
opt_Schlüsselwörter	Die optionalen Schlüsselwörter, die Sie beim Abrufen eines Solaris Flash-Archivs von einem HTTP-Server verwenden können.

TABELLE 8–3 Optionale Schlüsselwörter für `archive_location` als HTTP

Schlüsselwort	Definition des Werts
`auth basic` *Benutzername Passwort*	Befindet sich das Archiv auf einem passwortgeschützten HTTP-Server, müssen Sie den Benutzernamen und das Passwort für den Zugriff auf den HTTP-Server in die Profildatei aufnehmen. **Hinweis –** Der Einsatz dieser Authentisierungsmethode in einen Profil, das im Rahmen einer benutzerdefinierten JumpStart-Installation eingesetzt werden soll, stellt ein Sicherheitsrisiko dar. Nicht dazu berechtigte Benutzer könnten Zugriff auf das Profil erhalten, das das Passwort enthält.

TABELLE 8–3 Optionale Schlüsselwörter für `archive_location` als HTTP *(Fortsetzung)*

Schlüsselwort	Definition des Werts
`timeout` *Min*	Mit dem Schlüsselwort `timeout` können Sie in Minuten angeben, wie lange maximal keine Daten vom HTTP-Server eingehen dürfen, bevor die Verbindung beendet, wieder hergestellt und an dem Punkt, an dem die Zeitüberschreitung aufgetreten ist, wieder aufgenommen wird. Setzen Sie `timeout` auf den Wert 0 (Null), wird die Verbindung bei Inaktivität nicht neu hergestellt. ■ Erfolgt eine Neuverbindung nach Zeitüberschreitung, versuchen die Solaris Flash-Installationsdienstprogramme die Installation an der zuletzt bekannten Position im Archiv fortzusetzen. Sollten die Solaris Flash-Installationsdienstprogramme die Installation nicht an der zuletzt bekannten Position fortsetzen können, beginnt der Abruf wieder am Anfang des Archivs und die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen. ■ Sollte eine Verbindung nach einer Zeitüberschreitung neu aufgebaut werden, während gerade ein Package installiert wird, erfolgt ein neuer Versuch beim Anfang des Packages. Die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen.
`proxy` *Host:Port*	Das Schlüsselwort `proxy` ermöglicht es, einen Proxy-Host und einen Proxy-Port festzulegen. Mithilfe eines Proxy-Hosts können Sie Solaris Flash-Archive jenseits einer Firewall abrufen. Wenn Sie das Schlüsselwort `proxy` verwenden, müssen Sie auch einen Proxy-Port angeben.

BEISPIEL 8–2 HTTP- oder HTTPS-Server

```
archive_location http://silver/archives/usrarchive.flar timeout 5
```

Beispiel für das Schlüsselwort `auth basic` *Benutzername Passwort*:

```
archive_location http://silver/archives/usrarchive.flar timeout 5 user1 secret
```

FTP-Server

Ist das Archiv auf einem FTP-Server gespeichert, verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

`archive_location ftp://`*Benutzername Passwort*`@`*server_name*`:`*port*`/`*path*`/`*filename* *Optionale Schlüsselwörter*

Benutzer:Passwort	Der Benutzername und das Passwort, die Sie für den Zugriff auf den FTP-Server in der Profildatei benötigen.
Server	Der Name des Servers, auf dem Sie das Archiv gespeichert haben.
Port	Ein optionaler Port (Anschluss). *Port* kann eine Portnummer oder der Name eines TCP-Dienstes mit einer Portnummer sein, die bei der Ausführung ermittelt wird.

	Wenn Sie keinen Port angeben, verwenden die Solaris Flash-Installationsdienstprogramme den FTP-Standardport, Nummer 21.
Pfad	Die Speicherposition des vom angegebenen Server abzurufenden Archivs. Enthält der Pfad $HOST, so ersetzen die Solaris Flash-Installationsdienstprogramme $HOST durch den Namen des Klonsystems, das gerade installiert wird.
Dateiname	Der Name der Solaris Flash-Archivdatei.
opt_Schlüsselwörter	Die optionalen Schlüsselwörter, die Sie angeben können, wenn Sie ein Solaris Flash-Archiv von einem FTP-Server abrufen.

TABELLE 8–4 Optionale Schlüsselwörter für `archive_location` als FTP

Schlüsselwort	Definition des Werts
`timeout` *Min*	Mit dem Schlüsselwort `timeout` können Sie in Minuten angeben, wie lange maximal keine Daten vom HTTP-Server eingehen dürfen, bevor die Verbindung beendet, wieder hergestellt und an dem Punkt, an dem die Zeitüberschreitung aufgetreten ist, wieder aufgenommen wird. Setzen Sie `timeout` auf den Wert 0 (Null), wird die Verbindung bei Inaktivität nicht neu hergestellt. ■ Erfolgt eine Neuverbindung nach Zeitüberschreitung, versuchen die Solaris Flash-Installationsdienstprogramme die Installation an der zuletzt bekannten Position im Archiv fortzusetzen. Sollten die Solaris Flash-Installationsdienstprogramme die Installation nicht an der zuletzt bekannten Position fortsetzen können, beginnt der Abruf wieder am Anfang des Archivs und die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen. ■ Sollte eine Verbindung nach einer Zeitüberschreitung neu aufgebaut werden, während gerade ein Package installiert wird, erfolgt ein neuer Versuch beim Anfang des Packages. Die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen.
`proxy` *Host:Port*	Das Schlüsselwort `proxy` ermöglicht es, einen Proxy-Host und einen Proxy-Port festzulegen. Mithilfe eines Proxy-Hosts können Sie Solaris Flash-Archive jenseits einer Firewall abrufen. Wenn Sie das Schlüsselwort `proxy` verwenden, müssen Sie auch einen Proxy-Port angeben.

BEISPIEL 8–3 FTP-Server

```
archive_location ftp://user1:secret@silver/archives/usrarchive.flar timeout 5
```

Lokales Bandlaufwerk

Ist ein Archiv auf einem Band gespeichert, verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

```
archive_location local_tape Gerät Position
```

Gerät Der Name des Bandlaufwerks, auf dem Sie das Solaris Flash-Archiv gespeichert haben. Handelt es sich bei dem Gerätenamen um einen standardisierten Pfad, rufen die Solaris Flash-Installationsdienstprogramme das Archiv vom Pfad zum Geräteknoten ab. Wenn Sie einen Gerätenamen angeben, der kein standardisierter Pfad ist, dann fügen die Solaris Flash-Installationsdienstprogramme den Abschnitt `/dev/rmt/` an den Pfad an.

Position Gibt die genaue Speicherposition des Archivs auf dem Band an. Wenn Sie keine Position angeben, rufen die Solaris Flash-Installationsdienstprogramme das Archiv von der aktuellen Position im Bandlaufwerk ab. Durch Angabe einer *Position* haben Sie die Möglichkeit, ein Begin-Skript oder eine `sysidcfg`-Datei vor das Archiv auf dem Band einzufügen.

BEISPIEL 8–4 Lokales Bandlaufwerk

```
archive_location local_tape /dev/rmt/0n 5

archive_location local_tape 0n 5
```

Lokales Gerät

Ein auf einem dateisystemorientierten Direktspeichermedium wie z. B. einer Diskette oder einer DVD gespeichertes Solaris Flash-Archiv können Sie von einem lokalen Gerät abrufen. Verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

Hinweis – Mit der Syntax für lokale Bandlaufwerke können Sie Archive von streamorientierten Medien, wie z. B. Bändern, abrufen.

`archive_location local_device` *Gerät Pfad/Dateiname Dateisystemtyp*

Gerät Der Name des Laufwerks, auf dem Sie das Solaris Flash-Archiv gespeichert haben. Handelt es sich beim Gerätenamen um einen standardisierten Pfad, wird das Gerät direkt eingehängt. Wenn Sie einen Gerätenamen angeben, der kein standardisierter Pfad ist, dann fügen die Solaris Flash-Installationsdienstprogramme den Abschnitt `/dev/dsk/` an den Pfad an.

Pfad Der Pfad zum Solaris Flash-Archiv relativ zum Root des Dateisystems auf dem angegebenen Gerät. Enthält der Pfad $HOST, so ersetzen die Solaris Flash-Installationsdienstprogramme $HOST durch den Namen des Klonsystems, das gerade installiert wird.

Dateiname Der Name der Solaris Flash-Archivdatei.

Dateisystemtyp Gibt den Dateisystemtyp auf dem Gerät an. Wenn Sie keinen Dateisystemtyp angeben, versuchen die Solaris Flash-Installationsdienstprogramme, ein UFS-Dateisystem einzuhängen. Sollte der UFS-Einhängevorgang fehlschlagen, versuchen die Solaris Flash-Installationsdienstprogramme, ein HSFS-Dateisystem einzuhängen.

BEISPIEL 8–5 Lokales Gerät

Um ein Archiv von einer lokalen Festplatte abzurufen, die als ein UFS-Dateisystem formatiert ist, verwenden Sie den folgenden Befehl:

```
archive_location local_device c0t0d0s0 /archives/$HOST
```

Um ein Archiv von einer lokalen CD-ROM mit einem HSFS-Dateisystem abzurufen, verwenden Sie den folgenden Befehl:

```
archive_location local_device c0t0d0s0 /archives/usrarchive
```

Lokale Datei

Sie können ein in dem Miniroot, von welchem Sie das Klonsystem gebootet haben, gespeichertes Archiv als lokale Datei abrufen. Wenn Sie eine benutzerdefinierte JumpStart-Installation vornehmen, booten Sie das System von einer DVD, einer CD-ROM oder einer NFS-basierten Miniroot. Die Installationssoftware wird von dieser Miniroot geladen und ausgeführt. Folglich ist ein Solaris Flash-Archiv, das Sie auf der DVD, der CD-ROM oder in der NFS-basierten Miniroot gespeichert haben, als lokale Datei zugänglich. Verwenden Sie für das Schlüsselwort `archive_location` die folgende Syntax.

`archive_location local_file` *Pfad/Dateiname*

Pfad Der Speicherort des Archivs. Das System muss auf den Pfad wie auf eine lokale Datei zugreifen können, während das System von der Solaris Software-1-CD oder der Solaris Operating System-DVD aus gebootet wird. Auf `/net` oder andere automatisch eingehängte Verzeichnisse kann das System nicht zugreifen, während es von der Solaris Software-1-CD oder der Solaris Operating System-DVD aus gebootet wird.

Dateiname Der Name der Solaris Flash-Archivdatei.

BEISPIEL 8–6 Lokale Datei

```
archive_location local_file /archives/usrarchive
```

`backup_media`-Profilschlüsselwort

`backup_media` *Typ Pfad*

Das Schlüsselwort `backup_media` können Sie nur mit der Upgrade-Option verwenden, wenn eine Neuzuweisung von Festplattenspeicher erforderlich ist.

`backup_media` definiert die Medien, die zum Sichern von Dateisystemen verwendet werden, wenn während eines Upgrades aufgrund von unzureichendem Speicherplatz eine Neuzuweisung des Festplattenspeichers erfolgen muss. Wenn für die Sicherung mehrere Bänder oder Disketten benötigt werden, werden Sie während des Upgrades zum Einlegen der Bänder bzw. Disketten aufgefordert.

Gültige Werte für *Typ*	**Gültige Werte für** *Pfad*	**Spezifikation**
`local_tape`	`/dev/rmt/`*n*	Ein lokales Bandlaufwerk auf dem System, auf dem das Upgrade ausgeführt wird. *Pfad* muss der zeichenorientierte (raw) Gerätepfad für das Bandlaufwerk sein. *n* gibt die Nummer des Bandlaufwerks an.
`local_diskette`	`/dev/rdisketten`	Eine lokales Diskettenlaufwerk auf dem System, auf dem das Upgrade ausgeführt wird. *Pfad* muss der zeichenorientierte (raw) Gerätepfad für das Diskettenlaufwerk sein. *n* gibt die Nummer des Diskettenlaufwerks an. Die für die Sicherung verwendeten Disketten müssen formatiert sein.
`local_filesystem`	`/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z* */Dateisystem*	Ein lokales Dateisystem auf dem System, auf dem das Upgrade ausgeführt wird. Sie können kein lokales Dateisystem angeben, das beim Upgrade modifiziert wird. *Pfad* kann ein blockorientierter Gerätepfad für einen Festplatten-Speicherbereich sein. Die Angabe `t`*x* in `/dev/dsk/c` *w*`t`*x*`d` *y*`s`*z* wird möglicherweise nicht benötigt. Stattdessen können Sie als *Pfad* auch den absoluten Pfad zu einem Dateisystem eingeben, das von der Datei `/etc/vfstab` eingehängt wird.
`remote_filesystem`	*Host*`:`*/Dateisystem*	Ein NFS-Dateisystem auf einem Remote-System. *Pfad* muss den Namen oder die IP-Adresse des Remote-Systems (*Host*) und den absoluten Pfad zum NFS-Dateisystem (*Dateisystem*) enthalten. Auf das NFS-Dateisystem muss Lese-/Schreibzugriff bestehen.

Gültige Werte für *Typ*	Gültige Werte für *Pfad*	Spezifikation
`remote_system`	*Benutzer*`@`*Host*`:/`*Verzeichnis*	Ein Verzeichnis auf einem Remote-System, auf das über eine Remote Shell, `rsh`, Zugriff besteht. Das System, auf dem das Upgrade ausgeführt wird, muss über die Datei `.rhosts` des Remote-Systems Zugriff auf das Remote-System haben. *Pfad* muss den Namen des Remote-Systems (*Host*) und den absoluten Pfad zu dem Verzeichnis (*Verzeichnis*) enthalten. Wird keine Benutzer-Login-ID (*Benutzer*) angegeben, wird standardmäßig `root` verwendet.

BEISPIEL 8–7 `backup_media`-Profilschlüsselwort

```
backup_media local_tape /dev/rmt/0

backup_media local_diskette /dev/rdiskette1

backup_media local_filesystem /dev/dsk/c0t3d0s4

backup_media local_filesystem /export

backup_media remote_filesystem system1:/export/temp

backup_media remote_system user1@system1:/export/temp
```

`boot_device`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `boot_device` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`boot_device` *Gerät EEPROM*

`boot_device` gibt das Gerät an, auf dem das JumpStart-Programm das Root-Dateisystem (`/`) und das Boot-Gerät des Systems installiert. `boot_device` muss allen `filesys`-Schlüsselwörtern entsprechen, in denen das Root-Dateisystem (`/`) sowie das Schlüsselwort `root_device` angegeben ist.

Wird das Schlüsselwort `boot_device` nicht in einem Profil angegeben, so wird während der Installation der folgende Vorgabewert für `boot_device` verwendet:

```
boot_device any update
```

Gerät Verwenden Sie hier einen der folgenden Werte:

SPARC: c*w* t*x*d*y* s*z* oder c*x* d*y*s*z*	Der·Festplatten-Speicherbereich, in welchen das JumpStart-Programm das Root-Dateisystem (`/`) stellt, zum Beispiel `c0t0d0s0`.
x86: c*w* t*x*d*y* oder c*x*d*y*	Die Festplatte, auf die das JumpStart-Programm das Root-Dateisystem (`/`) stellt, zum Beispiel `c0d0`.
`existing`	Das JumpStart-Programm stellt das Root-Dateisystem (`/`) auf das vorhandene Boot-Gerät des Systems.
`any`	Das JumpStart-Programm wählt aus, wohin das Root-Dateisystem (`/`) gestellt wird. Dabei versucht das JumpStart-Programm, das vorhandene Boot-Gerät des Systems zu verwenden. Ist dies nicht möglich, wählt das JumpStart-Programm ein anderes Boot-Gerät aus.

EEPROM Gibt an, ob das EEPROM des Systems aktualisiert oder beibehalten wird.

Sie können mit dem *eeprom*-Wert außerdem das EEPROM des Systems ändern, wenn sich das aktuelle Boot-Gerät des Systems ändert. Indem Sie das EEPROM des Systems ändern, kann es automatisch vom neuen Boot-Gerät gestartet werden.

Hinweis – x86: Den Wert `preserve` müssen Sie angeben.

`update`	Das JumpStart-Programm ändert im EEPROM des Systems den Boot-Gerätewert, so dass das installierte System automatisch von dem neuen Boot-Gerät gestartet wird.
`preserve`	Der Boot-Gerätewert im EEPROM des Systems wird nicht geändert. Wenn Sie ein neues Boot-Gerät festlegen, ohne das EEPROM des Systems zu aktualisieren, müssen Sie das EEPROM des Systems von Hand ändern, so dass das System automatisch vom neuen Boot-Gerät gestartet werden kann.

BEISPIEL 8–8 `boot_device`-Profilschlüsselwort

```
boot_device c0t0d0s2 update
```

`bootenv`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `bootenv` kann für ein UFS-Dateisystem oder die Installation eines ZFS-Root-Pools verwendet werden. Die Syntax ist für ZFS-Installationen unterschiedlich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Eine Beschreibung zur Verwendung des Schlüsselworts `bootenv` für die Installation eines ZFS-Root-Pools finden Sie in „JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick)“ auf Seite 177.

`bootenv createbe bename` *neuer_BU-Name* `filesystem` *Einhängepunkt*`:`*Gerät*`:`*Dateisystemoptionen* `[filesystem...]`

Mit dem Schlüsselwort `bootenv createbe` können Sie gleichzeitig mit der Installation von Solaris eine leere, inaktive Boot-Umgebung erstellen. Sie müssen zumindest das Root-Dateisystem (`/`) anlegen. Die Speicherbereiche werden für die angegebenen Dateisysteme reserviert, es werden aber keine Dateisysteme kopiert. Die Boot-Umgebung wird zwar benannt, aber noch nicht tatsächlich erzeugt. Dies geschieht erst mit der Installation eines Solaris Flash-Archivs. Wenn Sie in der leeren Boot-Umgebung ein Archiv installieren, werden auf den reservierten Speicherbereichen Dateisysteme angelegt. Im Folgenden sind die möglichen Werte für *BU-Name* und *Dateisystem* aufgeführt.

`bename` *Neuer_BU-Name*
: `bename` legt den Namen der neu anzulegenden Boot-Umgebung fest. *Neuer_BU-Name* darf maximal 30 Zeichen lang sein und darf nur alphanumerische Zeichen enthalten. Multibyte-Zeichen sind nicht zulässig. Der Name muss auf dem System einmalig sein.

`filesystem` *Einhängepunkt*`:`*Gerät*`:`*DS-Optionen*
: `filesystem` gibt an, welche und wie viele Dateisysteme in der neuen Boot-Umgebung angelegt werden. Es muss zumindest ein Speicherbereich definiert werden, der·das Root-Dateisystem (`/`) enthält. Die Dateisysteme können sich auf derselben Festplatte befinden oder sich über mehrer Festplatten erstrecken.

  - Für *Einhängepunkt* können Sie einen beliebigen gültigen Einhängepunkt oder – (Bindestrich) angeben. Letzteres gibt an, dass es sich um einen Swap-Speicherbereich handelt.

- Das *Gerät* muss beim ersten Booten des installierten Betriebssystems verfügbar sein. Dieses Gerät hat keinen Bezug zu besonderen JumpStart-Speichergeräten wie z. B. *free*. Bei dem Gerät darf es sich nicht um ein Solaris Volume Manager-Volume oder ein Veritas Volume Manager-Volume handeln. *Gerät* ist der Name eines Festplattengeräts in der Form `/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z*.
- *DS-Optionen* kann einen der folgenden Werte erhalten:
  - `ufs`, ein UFS-Dateisystem
  - `swap`, ein Swap-Dateisystem. Der Swap-Einhängepunkt muss ein Bindestrich (–) sein.

Für ein Beispielprofil und Hintergrundinformationen zu diesem Schlüsselwort schlagen Sie bitte an folgenden Stellen nach:

Beispielprofil	Beispiel 3–11
Hintergrundinformationen zu Solaris Live Upgrade zum Erstellen, Aktualisieren und Aktivieren inaktiver Boot-Umgebungen	Kapitel 2, „Solaris Live Upgrade (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*
Hintergrundinformationen zur Arbeit mit Solaris Flash-Archiven	Kapitel 1, „Solaris Flash (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Flash-Archive (Erstellung und Installation)*

`client_arch`-Profilschlüsselwort

`client_arch` *karch_Wert* `...`

`client_arch` gibt an, dass der Betriebssystemserver eine andere Plattformgruppe als die vom Server selbst verwendete unterstützen soll. Wenn Sie `client_arch` im Profil nicht angeben, muss ein etwaiger Diskless-Client, der von dem Betriebssystemserver abhängt, dieselbe Plattformgruppe wie der Server enthalten. Sie müssen alle Plattformgruppen angeben, die vom Betriebssystemserver unterstützt werden sollen.

Gültige Werte für *karch_Wert* sind `sun4u` und `i86pc`. Eine ausführliche Liste der Plattformnamen und der verschiedenen Systeme finden Sie im *Solaris Handbuch zur Hardware-Plattform von Sun* unter·`http://docs.sun.com`.

Hinweis – Sie können `client_arch` nur verwenden, wenn für `system_type` der Wert `server` angegeben wird.

`client_root`-Profilschlüsselwort

`client_root` *Root-Größe*

`client_root` definiert, wie viel Root-Speicherplatz in MB (*Root-Größe*) den einzelnen Clients zugewiesen wird. Wenn Sie in einem Serverprofil `client_root` nicht angeben, weist die Installationssoftware 15 MB Root-Speicher pro Client zu. Die Größe des Client-Root-Bereichs wird in Kombination mit dem Schlüsselwort `num_clients` verwendet, um zu ermitteln, wie viel Speicherplatz für das Dateisystem `/export/root` reserviert werden muss.

Hinweis – Sie können `client_root` nur verwenden, wenn für `system_type` der Wert `server` angegeben wird.

`client_swap`-Profilschlüsselwort

`client_swap` *Swap-Größe*

`client_swap` definiert die Größe des Swap-Bereichs in MB (*Swap-Größe*), der den einzelnen Diskless-Clients zugewiesen wird. Wenn Sie `client_swap` im Profil nicht angeben, werden standardmäßig 32 MB für den Swap-Bereich zugewiesen.

Hinweis – Sie können `client_swap` nur verwenden, wenn für `system_type` der Wert `server` angegeben wird.

BEISPIEL 8–9 `client_swap`-Profilschlüsselwort

Im folgenden Beispiel wird festgelegt, dass alle Diskless-Clients einen Swap-Bereich mit 64 MB aufweisen sollen.

```
client_swap 64
```

So wird die `swap`-Größe festgelegt

Wenn ein Profil die `swap`-Größe nicht angibt, legt das JumpStart-Programm die Größe des Swap-Bereichs auf der Grundlage des physischen Hauptspeicher des Systems fest. Tabelle 8–5 zeigt, wie die Größe des `Swap`-Bereichs während einer benutzerdefinierten JumpStart-Installation festgelegt wird.

TABELLE 8–5 Festlegen der `swap`-Größe

Physischer Hauptspeicher (in MB)	Swap-Bereich (in MB)
16 – 64	32
64 – 128	64
128 – 512	128

TABELLE 8–5 Festlegen der swap-Größe *(Fortsetzung)*

Physischer Hauptspeicher (in MB)	Swap-Bereich (in MB)
Über 512	256

Das JumpStart-Programm legt als `swap`-Größe nicht mehr als 20 Prozent der Größe der Festplatte fest, auf der sich `swap` befindet, es sei denn, die Festplatte enthält nach dem Festlegen des Layouts der anderen Dateisysteme noch freien Speicherplatz. Wenn freier Speicherplatz vorhanden ist, weist das JumpStart-Programm diesen freien Speicherplatz – und, sofern möglich, den in der `Tabelle&;8–5` angezeigten Bereich – zum Tabelle 8–5-Bereich zu .

Hinweis – Der physische Hauptspeicher und der Swap-Bereich müssen zusammen mindestens 32 MB aufweisen.

`cluster`-Schlüsselwort (Hinzufügen von Softwaregruppen) (UFS und ZFS)

Hinweis – Das Schlüsselwort `cluster` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zur Verwendung des Schlüsselworts `cluster` in einem Profil für die Installation eines ZFS-Root-Pools finden Sie in „JumpStart-Profilbeispiele für ein ZFS-Root-Pool“ auf Seite 173.

`cluster` *Gruppenname*

`cluster` gibt die zu dem System hinzuzufügende Softwaregruppe an.

Eine Softwaregruppe ist ein Metacluster, der einen Satz von Clustern und Paketen enthält. Die Softwaregruppe wird mittels des Schlüsselworts `cluster` und der Variable *Gruppenname* installiert. Das cluster-Schlüsselwort kann nur bei einer Erstinstallation verwendet werden. Das Schlüsselwort `cluster` verweist auf Metacluster aus der Datei `clustertoc(4)`.

Ein Cluster ist ein Satz von Paketen, der einen Namen der Form SUNW *Name* trägt. Ein Cluster wird mittels des Schlüsselworts `cluster` und der Variable *Clustername* installiert. Cluster können bei einer Erstinstallation oder einem Upgrade zu einer Softwaregruppe (d. h. einem Metacluster) hinzugefügt oder daraus entfernt werden.

Der *Gruppenname* für die einzelnen Softwaregruppen ist in der folgenden Tabelle aufgeführt.

Softwaregruppe	*Gruppenname*
Softwaregruppe mit eingeschränkter Netzwerkunterstützung	`SUNWCrnet`
Softwaregruppe zur Hauptsystemunterstützung	`SUNWCreq`
Solaris-Softwaregruppe für Endbenutzer	`SUNWCuser`
Softwaregruppe für Entwickler	`SUNWCprog`
Gesamte Solaris-Softwaregruppe	`SUNWCall`
Gesamte Solaris-Softwaregruppe plus OEM-Unterstützung	`SUNWCXall`

Es gelten die folgenden Einschränkungen:

- Sie können in einem Profil nur eine Softwaregruppe angeben.
- Die Softwaregruppe muss vor anderen `cluster`- und `package`-Einträgen angegeben werden.
- Wenn Sie im Profil nicht mit `cluster` eine Softwaregruppe angeben, wird die Softwaregruppe End User, `SUNWCuser`, auf dem System installiert.

Weitere Informationen zu Softwaregruppen finden Sie unter „Empfohlener Festplattenspeicher für Softwaregruppen“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

`cluster`-Schlüsselwort (Hinzufügen bzw. Löschen von Clustern) (UFS und ZFS)

`cluster` *Clustername add_delete_switch*

Hinweis – Das Schlüsselwort `cluster` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zur Verwendung des Schlüsselworts `cluster` in einem Profil für die Installation eines ZFS-Root-Pools finden Sie in „JumpStart-Profilbeispiele für ein ZFS-Root-Pool“ auf Seite 173.

`cluster` legt fest, ob der auf dem System zu installierenden Softwaregruppe ein Cluster hinzugefügt oder daraus gelöscht wird.

Cluster-Name	Der Name des Clusters. Dieser muss das Format `SUNWC`*Name* aufweisen.
hinzufügen_oder_löschen	Dieses optionale Schlüsselwort gibt an, ob das angegebene Cluster hinzugefügt oder gelöscht werden soll. Als Werte sind `add` (Hinzufügen) und `delete` (Löschen) möglich. Wenn Sie weder `add` noch `delete` angeben, gilt standardmäßig `add`.

Wenn Sie `cluster` während eines Upgrades verwenden, gelten die folgenden Bedingungen:

- Alle bereits auf dem System vorhandenen Cluster werden aktualisiert.
- Wenn Sie *Clustername* `add` angeben und *Clustername* nicht auf dem System installiert ist, wird das Cluster installiert.
- Wenn Sie *Clustername* `delete` angeben und *Clustername* auf dem System installiert ist, wird das Package gelöscht, *bevor* das Upgrade beginnt.

Hinweis – Eine Softwaregruppe ist ein Metacluster, der einen Satz von Clustern und Paketen enthält. Die Softwaregruppe wird mittels des Schlüsselworts `cluster` und der Variable *Gruppenname* installiert. Das cluster-Schlüsselwort kann nur bei einer Erstinstallation verwendet werden. Das Schlüsselwort `cluster` verweist auf Metacluster aus der Datei `clustertoc(4)`.

Ein Cluster ist ein Satz von Paketen. Cluster können zu Softwaregruppen (Metaclustern) zusammengefasst werden. Clusternamen haben immer die Form SUNW<Name>. Ein Cluster wird mittels des Schlüsselworts `cluster` und der Variable *Clustername* installiert. Cluster können bei einer Erstinstallation oder einem Upgrade zu einer Softwaregruppe (d. h. einem Metacluster) hinzugefügt oder daraus entfernt werden.

-Profilschlüsselwort`dontuse` (UFS und ZFS)

Hinweis – Das Schlüsselwort `cluster` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`dontuse` *Festplattenname* `...`

Standardmäßig verwendet das JumpStart-Programm alle betriebsbereiten Festplatten im System, wenn Sie `partitioning default` angeben. Mit `dontuse` können Sie Festplatten festlegen, die vom JumpStart-Programm nicht verwendet werden sollen. *Festplattenname* muss das Format c*x*t *y*d*z* oder c *y*d*z* aufweisen, zum Beispiel `c0t0d0`.

Hinweis – Sie können die Schlüsselwörter `dontuse` und `usedisk` nicht in demselben Profil verwenden.

---

x86: `fdisk`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `fdisk` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`fdisk` *Festplattenname Typ Größe*

`fdisk` definiert, wie die `fdisk`-Partitionen auf einem x86-basierten System konfiguriert werden. Sie können `fdisk` mehr als einmal angeben. Bei der Partitionierung eines x86-basierten Systems mit `fdisk` geschieht Folgendes:

- Alle `fdisk`-Partitionen auf der Festplatte werden beibehalten, es sei denn, Sie löschen die Partitionen mit dem Schlüsselwort `fdisk`, indem Sie als *Größe* den Wert `delete` oder `0` angeben. Außerdem werden alle vorhandenen `fdisk`-Partitionen gelöscht, wenn *Größe* auf `all` gesetzt wird.
- Eine Solaris-`fdisk`-Partition, die ein Root-Dateisystem (`/`) enthält, wird immer zur aktiven Partition auf der Festplatte.

  Hinweis – Das System bootet standardmäßig von der aktiven Partition.

- Wenn das Schlüsselwort `fdisk` im Profil nicht angegeben wird, gilt das folgende `fdisk`-Schlüsselwort bei der Installation standardmäßig:

  ```
  fdisk all solaris maxfree
  ```

- `fdisk`-Einträge werden in der Reihenfolge verarbeitet, in der sie im Profil aufgeführt sind.

Festplattenname

Geben Sie mithilfe der folgenden Werte an, wo die `fdisk`-Partition erstellt oder gelöscht werden soll:

- c*x*t*y*d*z* oder c*y*d*z* – Eine bestimmte Festplatte, zum Beispiel `c0t3d0`.
- `rootdisk` – Die Variable, den den Wert der Root-Festplatte des Systems enthält, auf der die Installation stattfindet. Die Root-Festplatte wird vom JumpStart-Programm wie unter „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165 beschrieben ermittelt.
- `all` – Alle ausgewählten Festplatten.

Typ

Geben Sie mithilfe der folgenden Werte den Typ der `fdisk`-Partition an, die auf einer bestimmte Festplatte erstellt oder gelöscht werden soll:

- `solaris` – Eine Solaris-`fdisk`-Partition (Typ `SUNIXOS fdisk`).
- `dosprimary` – Ein Alias für primäre DOS-`fdisk`-Partitionen, nicht für `fdisk`-Partitionen, die erweitert oder für Daten-DOS reserviert sind. Wenn Sie `fdisk`-Partitionen löschen, indem Sie für *Größe* den Wert `delete` angeben, ist `dosprimary` ein Alias für die `fdisk`-Typen DOSHUGE, DOSOS12 und DOSOS16. Wenn Sie eine `fdisk`-Partition erstellen, ist `dosprimary` ein Alias für die DOSHUGE-`fdisk`-Partition.
- *DDD* – Eine `fdisk`-Partition als ganzzahliger Wert. *DDD* ist eine ganze Zahl von 1 bis 255 (jeweils einschließlich).

  Hinweis – Sie können diesen Wert nur angeben, wenn für *Größe* der Wert `delete` gilt.

- 0x*HH* – Eine `fdisk`-Partition als Hexadezimalwert. *HH* ist ein Hexadezimalwert zwischen 01 und FF.

  Hinweis – Sie können diesen Wert nur angeben, wenn für *Größe* der Wert `delete` gilt.

Die folgende Tabelle zeigt die ganzzahligen und Hexadezimalwerte für einige `fdisk`-Typen.

`fdisk`-**Typ**	*DDD*	*HH*
DOSOS12	`1`	`01`
PCIXOS	`2`	`02`
DOSOS16	`4`	`04`

`fdisk`-**Typ**	*DDD*	*HH*
EXTDOS	`5`	`05`
DOSHUGE	`6`	`06`
DOSDATA	`86`	`56`
OTHEROS	`98`	`62`
UNIXOS	`99`	`63`

Größe

Verwenden Sie hier einen der folgenden Werte:

- *DDD* – Eine `fdisk`-Partition der Größe *DDD* in MB wird auf der angegebenen Festplatte erstellt. *DDD* muss eine Ganzzahl sein und das JumpStart-Programm rundet die Zahl automatisch auf die nächste Zylindergrenze auf. Der Wert `0` hat dieselbe Funktion wie der Wert `delete`.
- `all` – Auf der gesamten Festplatte wird eine `fdisk`-Partition erstellt. Alle vorhandenen `fdisk`-Partitionen werden gelöscht.

  ---

  x86 nur – Der Wert `all` kann nur angegeben werden, wenn für *Typ* der Wert `solaris` gilt.

- `maxfree` – Eine `fdisk`-Partition wird in dem größten zusammenhängenden freien Speicherbereich auf der angegebenen Festplatte angelegt. Wenn auf der Festplatte bereits eine `fdisk`-Partition des angegebenen *Typs* vorhanden ist, wird die vorhandene `fdisk`-Partition verwendet. Auf der Festplatte wird in diesem Fall *keine* neue `fdisk`-Partition erstellt.

  ---

  x86 nur – Die Festplatte muss mindestens eine nicht benutzte `fdisk`-Partition aufweisen. Außerdem muss auf der Festplatte ausreichend freier Speicherplatz vorhanden sein, andernfalls schlägt die Installation fehl. Den Wert `maxfree` können Sie nur angeben, wenn für *Typ* der Wert `solaris` oder `dosprimary` gilt.

- `delete` – Alle `fdisk`-Partitionen des angegebenen *Typs* auf der angegebenen Festplatte werden gelöscht.

`filesys`-Profilschlüsselwort (Einhängen von Remote-Dateisystemen) (UFS und ZFS)

Hinweis – Das Schlüsselwort `filesys` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`filesys` *Server*:*Pfad Serveradresse Name_des_Einhängepunktes Einhängeoptionen*

Wenn Sie `filesys` mit den aufgeführten Werten verwenden, richtet das JumpStart-Programm das installierte System so ein, dass beim Booten des Systems automatisch Remote-Dateisysteme eingehängt werden. Sie können `filesys` mehr als einmal angeben.

Server — Der Name des Servers, auf dem sich das Remote-Dateisystem befindet, gefolgt von einem Doppelpunkt.

Pfad — Der Einhängepunktname des Remote-Dateisystems. Dies kann zum Beispiel `/usr` oder `/export/home` sein.

Serveradresse — Die IP-Adresse des Servers, der mit *Server*: *Pfad* angegeben ist. Wenn im Netzwerk kein Namen-Service läuft, können Sie mit dem Wert für *Serveradresse* den Host-Namen und die IP-Adresse des Servers in die Datei `/etc/hosts` einfügen. Wenn Sie die IP-Adresse des Servers nicht angeben, müssen Sie ein Minuszeichen (`-`) verwenden. Sie brauchen die IP-Adresse des Servers zum Beispiel nicht anzugeben, wenn im Netzwerk ein Namen-Service läuft.

Einhängepunktname — Der Name des Einhängepunkts, in dem das Remote-Dateisystem eingehängt werden soll.

Einhängeoptionen — Eine oder mehrere Einhängeoptionen. Dies ist mit der Option `-o` des Befehls `mount(1M)` identisch. Die Einhängeoptionen werden für den angegebenen *Einhängepunktnamen* zu dem `/etc/vfstab`-Eintrag hinzugefügt.

Hinweis – Wenn Sie mehr als eine Einhängeoption angeben müssen, trennen Sie die einzelnen Optionen durch Kommas ohne Leerzeichen (zum Beispiel `ro,quota`).

BEISPIEL 8–10 `filesys`-Profilschlüsselwort

```
filesys sherlock:/export/home/user2 - /home
```

`filesys`-Profilschlüsselwort (Erstellen lokaler Dateisysteme)

`filesys` *Speicherbereichsgröße Dateisystem Optionale_Parameter*

Wenn Sie `filesys` mit den aufgeführten Werten verwenden, erstellt das JumpStart-Programm während der Installation lokale Dateisysteme. Sie können `filesys` mehr als einmal angeben.

Speicherbereich Verwenden Sie hier einen der folgenden Werte:

`any`	Das JumpStart-Programm erstellt das Dateisystem auf einer beliebigen Festplatte.
	Hinweis – Sie können `any` nicht angeben, wenn für `Größe` der Wert `existing`, `all`, `free`, *Start*: *Größe* oder `ignore` gilt.
c*w* t*x*d*y* s*z* oder c*x* d*y*s*z*	Der·Festplatten-Speicherbereich, in den das JumpStart-Programm das Dateisystem stellt, zum Beispiel `c0t0d0s0` oder `c0d0s0`.
`rootdisk.s`*n*	Die Variable, die den Wert für die Root-Festplatte des Systems enthält. Diese wird vom JumpStart-Programm wie unter „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165 beschrieben ermittelt. Das Suffix s*n* gibt einen bestimmten Speicherbereich auf der Festplatte an.

Hinweis – Die Root-Festplatte wird vom JumpStart-Programm ermittelt und legt fest, wo das Betriebssystem installiert wird. Die rules-Datei verwendet das Probe-Schlüsselwort „`rootdisk`“, aber dieses Schlüsselwort wird anders als das Schlüsselwort „rootdisk“ im JumpStart-Profil verwendet. Sie können den Installationsort nicht mithilfe des Probe-Schlüsselworts `rootdisk` in der rules-Datei festlegen. Das Probe-Schlüsselwort `rootdisk` legt fest, von wo während der Installation gebootet wird. Siehe Tabelle 8–10.

Größe

Verwenden Sie hier einen der folgenden Werte:

num
: Die Größe des Dateisystem wird als *numerischer_Wert* in MB angegeben.

`existing`
: Die aktuelle Größe des vorhandenen Dateisystems wird verwendet.

  Hinweis – Wenn Sie den Wert `existing` verwenden, können Sie den Namen eines vorhandenen Speicherbereichs ändern, indem Sie für *Dateisystem* einen anderen *Einhängepunktnamen* angeben.

`auto`
: Die Größe des Dateisystems wird automatisch festgelegt, und zwar je nach der ausgewählten Software.

`all`
: Der·angegebene *Speicherbereich* verwendet die gesamte Festplatte für das Dateisystem. Wenn Sie den Wert `all` angeben, können keine weiteren Dateisysteme auf der entsprechenden Festplatte untergebracht werden.

`free`
: Der verbleibende freie Speicherplatz auf der Festplatte wird für das Dateisystem verwendet.

Hinweis – Wenn `free` als Wert für `filesys` verwendet wird, muss der `filesys`-Eintrag der letzte Eintrag im Profil sein.

Start:*Größe* — Das Dateisystem wird explizit partitioniert. *Start* gibt den Zylinder an, an dem der Speicherbereich beginnt. *Größe* gibt die Anzahl an Zylindern für den Speicherbereich an.

Dateisystem — Der Wert *Dateisystem* ist optional und wird verwendet, wenn für *Speicherbereich* der Wert `any` oder `c`*w*`t`*x*`d`*y*`s`*z* angegeben wird. Wenn *Dateisystem* nicht angegeben wird, gilt standardmäßig der Wert `unnamed`. Wenn `unnamed` verwendet wird, können Sie die *optionalen_Parameter* nicht angeben. Verwenden Sie hier einen der folgenden Werte:

- *Einhängepunktname* — Der Einhängepunktname des Dateisystems, zum Beispiel `/var`.
- `swap` — Der angegebene *Speicherbereich* wird als `swap`-Bereich verwendet.
- `overlap` — Der angegebene *Speicherbereich* wird als eine Darstellung eines Festplattenbereichs definiert. Der VTOC-Wert lautet V_BACKUP. Standardmäßig ist Speicherbereich 2 ein Overlap-Bereich, der eine Darstellung der gesamten Festplatte bildet.

  Hinweis – Sie können `overlap` nur angeben, wenn für *Größe* der Wert `existing`, `all` oder *Start*: *Größe* verwendet wird.

- `unnamed` — Der angegebene *Speicherbereich* wird als raw-Bereich definiert. In diesem Fall gibt es für *Speicherbereich* keinen Einhängepunktnamen. Wenn Sie *Dateisystem* nicht angeben, wird standardmäßig `unnamed` verwendet.
- `ignore` — Der angegebene *Speicherbereich* wird nicht verwendet und vom JumpStart-Programm

nicht erkannt. Mithilfe dieser Option können Sie festlegen, dass ein Dateisystem auf einer Festplatte während der Installation ignoriert wird. Das JumpStart-Programm erstellt ein neues Dateisystem auf der Festplatte mit demselben Namen. Sie können `ignore` nur verwenden, wenn `partitioning existing` angegeben wird.

optionale_Parameter Verwenden Sie hier einen der folgenden Werte:

`preserve` Das Dateisystem auf dem angegebenen *Speicherbereich* wird beibehalten.

Hinweis – `preserve` kann nur angegeben werden, wenn für *Größe* der Wert `existing` und für *Speicherbereich* der Wert `c`*w*`t`*x*`d`*y*`s`*z* gilt.

Einhängeoptionen Eine oder mehrere Einhängeoptionen. Dies ist mit der Option `-o` des Befehls `mount(1M)` identisch. Die Einhängeoptionen werden für den angegebenen *Einhängepunktnamen* zu dem `/etc/vfstab`-Eintrag hinzugefügt.

Hinweis – Wenn Sie mehr als eine Einhängeoption angeben müssen, trennen Sie die einzelnen Optionen durch Kommas ohne Leerzeichen (zum Beispiel `ro,quota`).

`filesys`-Profilschlüsselwort (Erstellen von RAID-1-Volumes)

`filesys mirror[:`*Name*`]`*slice* `[`*slice*`]` *Größe file_system Optionale_Parameter*

Durch Verwendung der Schlüsselwörter `filesys mirror` mit den aufgeführten Werten erzeugt das JumpStart-Programm die zum Erstellen eines gespiegelten Dateisystems erforderlichen RAID-1- und RAID-0-Volumes. Wenn Sie RAID-1-Volumes (Mirrors) für verschiedene Dateisysteme erstellen möchten, können Sie `filesys mirror` mehrmals angeben.

Hinweis – Das Schlüsselwort `filesys mirror` wird nur bei Neuinstallationen unterstützt.

Name	Dieses optionale Schlüsselwort ermöglicht es, das RAID-1-Volume (Mirror) zu benennen. Namen von RAID-1-Volumes (Mirrors) müssen mit dem Buchstaben "d" beginnen, auf den eine Zahl zwischen 0 und 127 folgt (z. B. `d100`). Wenn Sie für das RAID-1-Volume (Mirror) keinen Namen angeben, weist ihm das benutzerdefinierte JumpStart-Programm automatisch einen Namen zu. Richtlinien zur Benennung von RAID-1-Volumes (Mirrors) finden Sie unter „Voraussetzungen für RAID-Volume-Namen und Richtlinien für das benutzerdefinierte JumpStart-Verfahren sowie für Solaris Live Upgrade" in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.
Speicherbereich	Dieser Wert steht für den·Festplatten-Speicherbereich, auf dem das benutzerdefinierte JumpStart-Programm das zu spiegelnde Dateisystem anlegt. Für den Speicherbereichswert ist das Format c*w*t*x*d *y*s*z* zu verwenden, zum Beispiel `c0t0d0s0` oder `c0t0d0s5`. Das benutzerdefinierte JumpStart-Programm erzeugt ein RAID-0-Volume (Einzel-Speicherbereichsverkettung) auf dem Speicherbereich und ein RAID-1-Volume zum Spiegeln der Verkettung. Es können bis zu zwei Speicherbereiche für zwei RAID-0-Volumes angegeben werden.
Größe	Dieser Wert steht für die Größe des Dateisystems in MB.
Dateisystem	Mit diesem Wert geben Sie das zu spiegelnde Dateisystem an. Das benutzerdefinierte JumpStart-Programm erzeugt aus den angegebenen Speicherbereichen ein RAID-1-Volume und hängt dieses in das angegebene Dateisystem ein. Neben kritischen Dateisystemen wie Root (`/`), `/usr` und `/var` können Sie auch `swap` als Dateisystem angeben.
optionale_Parameter	Eine oder mehrere Einhängeoptionen. Dies ist mit der Option `-o` des Befehls `mount(1M)` identisch. Die Einhängeoptionen werden in den `/etc/vfstab`-Eintrag für das angegebene *Dateisystem* aufgenommen. Wenn Sie mehr als eine Einhängeoption angeben müssen, trennen Sie die einzelnen Optionen durch Kommas ohne Leerzeichen (zum Beispiel `ro,quota`).

Weitere Informationen zum Erstellen gespiegelter Dateisysteme während der Installation finden Sie in Kapitel 9, „Erstellen von RAID-1-Volumes (Mirrors) bei der Installation (Überblick)" in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

Das Profilschlüsselwort `forced_deployment` (Installation von Solaris Flash-Differenzarchiven)

`forced_deployment`

`forced_deployment` erzwingt die Installation eines Solaris Flash-Differenzarchivs auf einem Klonsystem, das die Software als von den Vorgaben abweichend erkennt.

Achtung – Bei der Verwendung von `forced_deployment` werden alle neuen Dateien gelöscht, um das Klon-System auf den richtigen Stand zu bringen. Wenn Sie nicht sicher sind, ob Dateien gelöscht werden sollen, verwenden Sie die Standardeinstellung. Dabei werden neue Dateien durch Unterbrechung der Installation geschützt.

`geo`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `geo` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`geo` *Region*

`geo` legt die regionale Sprachumgebung bzw. Sprachumgebungen fest, die auf einem System installiert oder beim Upgrade eines Systems hinzugefügt werden. *Region* steht dabei für ein geografisches Gebiet, das die zu installierenden Sprachumgebungen umfasst. In der folgenden Tabelle sind die Werte aufgeführt, die Sie für *Region* angeben können:

Wert	Beschreibung
`N_Africa`	Nordafrika, einschließlich Ägypten
`C_America`	Mittelamerika, einschließlich Costa Rica, El Salvador, Guatemala, Mexiko, Nicaragua, Panama
`N_America`	Nordamerika, einschließlich Kanada, USA
`S_America`	Südamerika, einschließlich Argentinien, Bolivien, Brasilien, Chile, Kolumbien, Ecuador, Paraguay, Peru, Uruguay, Venezuela

Wert	Beschreibung
`Asia`	Asien, einschließlich Japan, Republik Korea, Volksrepublik China, Taiwan, Thailand
`Ausi`	Australien und Ozeanien, einschließlich Australien, Neuseeland
`C_Europe`	Mitteleuropa, einschließlich Österreich, Tschechische Republik, Deutschland, Ungarn, Polen, Slowakei, Schweiz
`E_Europe`	Osteuropa, einschließlich Albanien, Bosnien, Bulgarien, Kroatien, Estland, Lettland, Litauen, Mazedonien, Rumänien, Russland, Serbien, Slowenien, Türkei
`N_Europe`	Nordeuropa, einschließlich Dänemark, Finnland, Island, Norwegen, Schweden
`S_Europe`	Südeuropa, einschließlich Griechenland, Italien, Portugal, Spanien
`W_Europe`	Westeuropa, einschließlich Belgien, Frankreich, Großbritannien, Irland, Niederlande
`M_East`	Naher Osten, einschließlich Israel

Eine vollständige Liste der Werte für einzelne Sprachumgebungen, die zusammen die oben genannten Sprachumgebungen für die übergeordnete geografische Region bilden, finden Sie im *International Language Environments Guide*.

Hinweis – Sie können für jede Sprachumgebung, die zu einem System hinzugefügt werden soll, ein `geo`-Schlüsselwort angeben.

`install_type`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `install_type` kann für ein UFS-Dateisystem oder die Installation eines ZFS-Root-Pools verwendet werden. Das Schlüsselwort kann nur ZFS-Installationen verwendet werden. Sie können nur die Option `intial_install` für eine ZFS-Installation verwenden.

- Wenn Sie ein UFS-Dateisystem auf ein ZFS-Root-Pool migrieren oder ein Upgrade eines ZFS-Root-Pools durchführen möchten, müssen Sie Solaris Live Upgrade verwenden. Siehe Kapitel 11, „Solaris Live Upgrade und ZFS (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.
- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Eine Beschreibung zur Verwendung des Schlüsselworts `install_type` für die Installation eines ZFS-Root-Pools finden Sie in „JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick)“ auf Seite 177.

`install_type` *initial_upgrade_flash_switch*

`install_type` definiert, ob das vorhandene Betriebssystem Solaris auf dem System gelöscht und ein neues installiert wird, ob ein Upgrade der vorhandenen Solaris-Installation ausgeführt wird oder ob ein Solaris Flash-Archiv auf dem System installiert wird.

Hinweis – Ein Profil muss das Schlüsselwort `install_type` enthalten und `install_type` muss das erste Profilschlüsselwort in allen Profilen sein.

Für *Neuinstallation_Upgrade_Flash* müssen Sie eine der folgenden Optionen verwenden:

`initial_install`	Gibt an, dass eine Erstinstallation (Neuinstallation) von Solaris vorgenommen wird.
`upgrade`	Gibt an, dass ein Upgrade von Solaris vorgenommen wird.
`flash_install`	Gibt an, dass ein Solaris Flash-Archiv installiert werden soll, das alle Dateien überschreibt.
`flash_update`	Gibt an, dass ein Solaris Flash-Differenzarchiv installiert werden soll, das nur die aufgeführten Dateien überschreibt.

Hinweis – Einige Profilschlüsselwörter können nur zusammen mit der Option `initial_install` verwendet werden. Einige Profilschlüsselwörter können nur zusammen mit der Option `upgrade` verwendet werden. Einige Profilschlüsselwörter können nur zusammen mit der Option `flash_install` verwendet werden.

`layout_constraint`-Profilschlüsselwort

`layout_constraint` *Speicherbereich Einschränkung Mindestgröße*

`layout_constraint` legt die Einschränkungen fest, die für Auto-Layout bei einem Dateisystem gelten, wenn Auto-Layout aufgrund von fehlendem Speicherplatz eine Neuzuweisung von Festplattenspeicher vornehmen muss.

Einschränkung	Beschreibung
Dieses Schlüsselwort wird nur in Verbindung mit der Upgrade-Option verwendet.	Das Schlüsselwort `layout_constraint` können Sie nur für die Upgrade-Option verwenden, wenn eine Neuzuweisung von Festplattenspeicher erforderlich ist.

Einschränkung	Beschreibung
Das Schlüsselwort `layout_constraint` wird nicht angegeben	In diesem Fall legt das JumpStart-Programm das Festplattenlayout folgendermaßen fest: ■ Dateisysteme, für die im Rahmen des Upgrades mehr Speicherplatz benötigt wird, werden als modifizierbar markiert. ■ Dateisysteme, die sich auf derselben Festplatte befinden wie ein Dateisystem, für das mehr Speicherplatz benötigt wird, und die über die Datei `/etc/vfstab` eingehängt werden, werden als modifizierbar markiert. ■ Die übrigen Dateisysteme werden als nicht modifizierbar markiert, da Auto-Layout diese nicht ändern kann.
Es wird mindestens ein Schlüsselwort `layout_constraint` angegeben	In diesem Fall legt das JumpStart-Programm das Festplattenlayout folgendermaßen fest: ■ Dateisysteme, für die im Rahmen des Upgrades mehr Speicherplatz benötigt wird, werden als modifizierbar markiert. ■ Dateisysteme, für die Sie ein `layout_constraint`-Schlüsselwort angeben, werden mit der angegebenen Einschränkung markiert. ■ Die übrigen Dateisysteme werden als nicht modifizierbar markiert.
Das Dateisystem ist nicht als modifizierbar markiert	Sie können keine Einschränkung für Dateisysteme festlegen, für die im Rahmen eines Upgrades mehr Speicherplatz benötigt wird, da diese Dateisysteme als modifizierbar markiert werden müssen. Sie können jedoch mit dem Schlüsselwort `layout_constraint` den Wert für die *Mindestgröße* von Dateisystemen ändern, für die im Rahmen eines Upgrades mehr Speicherplatz benötigt wird.
Dateisysteme benötigen mehr Speicherplatz für das Upgrade	Um die Neuzuweisung von Speicherplatz durch Auto-Layout zu erleichtern, sollten mehrere Dateisysteme als modifizierbar (changeable) oder verschiebbar (movable) markiert werden, insbesondere Dateisysteme, die sich auf derselben Festplatte befinden wie die Dateisysteme, für die im Rahmen eines Upgrades mehr Speicherplatz benötigt wird.

Speicherbereich Gibt den·Festplatten-Speicherbereich des Dateisystems an, auf den sich die Einschränkung beziehen soll. Sie müssen den·Festplatten-Speicherbereich im Format `c`*wtx*`d` *ysz* oder `c`*x* `d`*ysz* angeben.

Einschränkung Verwenden Sie für das angegebene Dateisystem eine der folgenden Einschränkungen:

`changeable` Auto-Layout kann das Dateisystem an eine andere Stelle verschieben und seine Größe ändern. Die Einschränkung `changeable` kann nur bei Dateisystemen angegeben

werden, die über die Datei `/etc/vfstab` eingehängt werden. Zum Ändern der Dateisystemgröße geben Sie den Wert *Mindestgröße* an.

Wenn Sie ein Dateisystem als modifizierbar markieren, nicht jedoch die *Mindestgröße* angeben, wird die Mindestgröße des Dateisystems auf 10 Prozent mehr als erforderlich festgelegt. Wenn die Mindestgröße für ein Dateisystem zum Beispiel 100 MB beträgt, gilt dann eine Mindestgröße von 110 MB. Wenn Sie *Mindestgröße* angeben, wird der verbleibende freie Speicherplatz, also die Originalgröße minus der Mindestgröße, für andere Dateisysteme verwendet.

`movable` Auto-Layout kann das Dateisystem in einen anderen Speicherbereich auf derselben Festplatte oder auf eine andere Festplatte verschieben. Die Dateisystemgröße bleibt unverändert.

`available` Auto-Layout kann für die Neuzuweisung von Speicherplatz den gesamten Speicherplatz im Dateisystem verwenden. Alle Daten im Dateisystem gehen verloren. Die Einschränkung `available` kann nur bei Dateisystemen angegeben werden, die nicht über die Datei `/etc/vfstab` eingehängt werden.

`collapse` Auto-Layout verschiebt das angegebene Dateisystem in das übergeordnete Dateisystem und führt diese zusammen. Mit der Option `collapse` können Sie die Anzahl der Dateisysteme auf einem System im Rahmen eines Upgrades verringern. Wenn auf einem System zum Beispiel die Dateisysteme `/usr` und `/usr/share` vorhanden sind und Sie für das Dateisystem `/usr/share` die Option "collapse" angeben, wird dieses Dateisystem in `/usr`, das übergeordnete Dateisystem, verschoben. Die Einschränkung `collapse` kann nur bei Dateisystemen angegeben werden, die über die Datei `/etc/vfstab` eingehängt werden.

Mindestgröße Gibt die Größe des Dateisystems nach der Neuzuweisung von Speicherplatz durch Auto-Layout an. Mit der Option *Mindestgröße* können Sie die Größe eines Dateisystems ändern. Das Dateisystem wird möglicherweise größer, wenn dem Dateisystem freier Speicherplatz zugewiesen wird. Das Dateisystem ist jedoch immer mindestens so groß wie angegeben. Der Wert *Mindestgröße* ist optional. Verwenden Sie diesen Wert nur, wenn Sie ein Dateisystem als modifizierbar markiert haben und die Mindestgröße nicht

kleiner als der Speicherplatz sein darf, den der Inhalt des Dateisystems bereits belegt.

BEISPIEL 8–11 `layout_constraint`-Profilschlüsselwort

```
layout_constraint c0t3d0s1 changeable 200

layout_constraint c0t3d0s4 movable

layout_constraint c0t3d1s3 available

layout_constraint c0t2d0s1 collapse
```

Das Profilschlüsselwort `local_customization` (Installation von Solaris Flash-Archiven)

`local_customization` *Lokales_Verzeichnis*

Vor der Installation eines Solaris Flash-Archivs auf einem Klonsystem können Sie benutzerdefinierte Skripten anwenden, um lokale Konfigurationen des Klonsystems beizubehalten. Das Schlüsselwort `local_customization` gibt das Verzeichnis an, in dem Sie diese Skripten gespeichert haben. *lokales_Verzeichnis* ist der Pfad zu dem Skript auf dem Klonsystem.

Informationen zu Predeployment- und Postdeployment-Skripten finden Sie unter „Erstellen von Anpassungsskripten“ in *Solaris 10 10/08 Installationshandbuch: Solaris Flash-Archive (Erstellung und Installation)*.

`locale`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `locale` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`locale` *Sprachumgebungsname*

Hinweis – Sie können `locale` bei einer Neuinstallation und bei einem Upgrade verwenden.

`locale` legt die Sprachumgebungs-Packages fest, die für den angegebenen *Sprachumgebungsnamen* installiert oder hinzugefügt werden. Die Werte für *Sprachumgebungsname* sind dieselben wie für die Umgebungsvariable `$LANG`. Eine Liste der Werte für gültige Sprachumgebungen finden Sie im *International Language Environments Guide*.

Beachten Sie Folgendes, wenn Sie das Schlüsselwort `locale` verwenden:

- Wenn Sie eine Standardsprachumgebung vorkonfiguriert haben, wird diese automatisch installiert. Standardmäßig werden die englischen Packages installiert.
- Sie können für jede Sprachumgebung, die zu einem System hinzugefügt werden soll, ein `locale`-Schlüsselwort angeben.

`metadb`-Profilschlüsselwort (Erstellen von Statusdatenbankreplikationen)

`metadb` *Speicherbereich* `[size` *Größ-in-Blöcken*`] [count` *Replika-Anzahl*`]`

Das Schlüsselwort `metadb` ermöglicht es, während der benutzerdefinierten JumpStart-Installation Solaris Volume Manager-Statusdatenbankreplikationen zu erstellen. Wenn Sie Statusdatenbankreplikationen auf verschiedenen Festplatten-Speicherbereichen erstellen möchten, können Sie das Schlüsselwort `metadb` im Profil mehrmals angeben.

Speicherbereich	Sie müssen den·Festplatten-Speicherbereich angeben, auf dem das benutzerdefinierte JumpStart-Programm die Statusdatenbankreplikation erzeugen soll. Der Wert für *Speicherbereich* muss das Format *cwtxdysz* aufweisen.
`size` *Größe_in_Blöcken*	Mit dem nicht obligatorischen Schlüsselwort `size` können Sie festlegen, wie groß in Blöcken die Statusdatenbankreplikation sein soll. Wenn Sie keinen Wert für `size` liefern, wendet das benutzerdefinierte JumpStart-Programm die Standardgröße von 8192 Blöcken auf die Statusdatenbankreplikation an.
`count` *Anz_Replikationen*	Indem Sie das optionale Schlüsselwort `count` in Ihrem Profil verwenden, können Sie festlegen, wie viele Statusdatenbankreplikationen erstellt werden sollen. Wenn Sie keinen Wert für `count` angeben, erstellt das benutzerdefinierte JumpStart-Programm standardmäßig drei Statusdatenbankreplikationen.

Weitere Informationen zum Erstellen von Solaris Volume Manager-Statusdatenbankreplikationen während der Installation finden Sie unter „Richtlinien und Voraussetzungen für Statusdatenbankreplikationen“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

Das Profilschlüsselwort `no_content_check` (Installation von Solaris Flash-Archiven)

`no_content_check`

Für die Installation von Klon-Systemen mit einem Solaris Flash-Differenzarchiv können Sie mit dem Schlüsselwort `no_content_check` angeben, dass keine Kontrolle der einzelnen Dateien durchgeführt wird. Bei dieser Kontrolle wird überprüft, ob das Klon-System eine exakte Kopie des Master-Systems ist. Sofern Sie nicht überzeugt sind, dass das Klon-System ein Duplikat des ursprünglichen Master-Systems ist, sollten Sie dieses Schlüsselwort nicht verwenden.

Achtung – Bei der Verwendung von `no_content_check` werden alle neuen Dateien gelöscht, um das Klonsystem auf den richtigen Stand zu bringen. Wenn Sie nicht sicher sind, ob Dateien gelöscht werden sollen, verwenden Sie die Standardeinstellung. Dabei werden neue Dateien durch Unterbrechung der Installation geschützt.

Informationen zur Installation von Solaris Flash-Differenzarchiven finden Sie in „So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor“ auf Seite 87.

Das Profilschlüsselwort `no_master_check` (Installation von Solaris Flash-Archiven)

`no_master_check`

Mit dem Schlüsselwort `no_master_check` lässt sich die bei der Installation eines Klonsystems mit einem Solaris Flash-Differenzarchiv normalerweise erfolgende Prüfung umgehen, durch die festgestellt wird, ob das Klonsystem aus dem Original-Mastersystem erzeugt wurde. Sofern Sie nicht überzeugt sind, dass das Klon-System ein Duplikat des ursprünglichen Master-Systems ist, sollten Sie dieses Schlüsselwort nicht verwenden.

Informationen zur Installation von Solaris Flash-Differenzarchiven finden Sie in „So bereiten Sie die Installation eines Solaris Flash-Archivs mit der benutzerdefinierten JumpStart-Installation vor“ auf Seite 87.

`num_clients`-Profilschlüsselwort

`num_clients` *Client_Anz*

Bei der Installation eines Servers wird Speicherplatz für die Root- (`/`) und `swap`-Dateisysteme aller Diskless-Clients zugewiesen. `num_clients` definiert die Anzahl an Diskless-Clients, *Anzahl_Clients*, die ein Server unterstützt. Wenn Sie `num_clients` im Profil nicht angeben, werden standardmäßig fünf Diskless-Clients zugewiesen.

Hinweis – Sie können `num_clients` nur verwenden, wenn für `system_type` der Wert `server` angegeben wird.

`package`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `package` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`package` *Package_Name* `[add [`*Abrufmethode Speicherort*`]| delete]`

Sie können `package` bei einer Neuinstallation und bei einem Upgrade verwenden. Das Schlüsselwort `package` bietet Ihnen folgende Möglichkeiten:

- Hinzufügen eines Packages aus der zu installierenden Solaris-Distribution zu einer Softwaregruppe
- Hinzufügen eines Packages aus einer externen Quelle zu einer Softwaregruppe
- Ausschließen bzw. Entfernen eines Pakets aus der Softwaregruppe, die installiert bzw. aktualisiert wird
- Hinzufügen eines Packages aus einer externen Quelle bei der Installation eines Solaris Flash-Archivs

Packagename	Gibt den Namen des Packages an. Dieser muss im Format `SUNW`*Name* vorliegen. Detaillierte Informationen zu Packages und deren Namen können Sie auf einem bereits installierten System mit dem Befehl `pkginfo -l` anzeigen lassen.
add | delete	Diese Parameter legen fest, ob das angegebene Package hinzugefügt oder gelöscht wird. Wenn Sie weder `add` noch `delete` angeben, gilt standardmäßig `add`.

Hinweis – Sie können mehrere Packages hinzufügen, indem Sie das Profil um weitere Package-Einträge erweitern. Eine erneute Angabe des Speicherorts ist dann nicht erforderlich. In diesem Fall verwendet das System automatisch den Speicherort des zuletzt angegebenen Packages für alle weiteren Packages.

[*Abrufmethode Speicherort*] Legt die Angaben fest, die beim Hinzufügen von Packages erforderlich sind, die nicht in der zu installierenden Solaris-Distribution enthalten sind. Je nach Speicherort des Packages müssen Sie unterschiedliche Werte für *Abrufmethode* und *Speicherort* angeben. In den folgenden Abschnitten werden die Werte, die Sie für *Abrufmethode* und *Position* angeben können, sowie Beispiele für die Verwendung des Schlüsselworts `archive_location` beschrieben.

NFS-Server

Wenn das fragliche Package auf einem NFS-Server gespeichert ist, verwenden Sie für `package` eine der folgenden Syntaxmöglichkeiten.

```
package Packagename add nfs Servername:/path [retry n]
package Packagename add nfs://Servername:/path [retry n]
```

Packagename Gibt den Namen des Packages an. Dieser muss im Format `SUNW`*Name* vorliegen. Detaillierte Informationen zu Packages und deren Namen können Sie auf einem bereits installierten System mit dem Befehl `pkginfo -l` anzeigen lassen.

Server Gibt den Namen des Servers an, auf dem das Package gespeichert ist.

Pfad Gibt den Ort des Package-Verzeichnisses auf dem angegebenen Server an. Wenn der Pfad den Platzhalter $HOST enthält, wird dieser durch den Namen des zu installierenden Host-Systems ersetzt.

retry n Optionales Schlüsselwort. Der Installationsvorgang versucht hiermit maximal *n* Mal, das Verzeichnis einzuhängen.

BEISPIEL 8–12 Hinzufügen eines Packages über NFS

In diesem Beispiel wird mit dem Profilschlüsselwort `package` das Package SUNWnew vom NFS-Speicherort `nfs://golden/packages/Solaris_10/` hinzugefügt. Sollte das NFS-Einhängen fehlschlagen, werden maximal fünf Versuche unternommen.

```
package SUNWnew add nfs golden:/packages/Solaris_10 retry 5
```

HTTP-Server

Wenn das fragliche Package auf einem HTTP-Server gespeichert ist, verwenden Sie für `package` eine der folgenden Syntaxmöglichkeiten.

`package` *Packagename* `add http://`*server_name[:port] path Optionale_Schlüsselwörter*
`Package` *Packagename* `add http` *Servername[:Port] Pfad Optionale_Schlüsselwörter*

Packagename	Gibt den Namen des Packages an. Dieser muss im Format `SUNW`*Name* vorliegen. Detaillierte Informationen zu Packages und deren Namen können Sie auf einem bereits installierten System mit dem Befehl `pkginfo -l` anzeigen lassen.
Server	Gibt den Namen des Servers an, auf dem das Package gespeichert ist.
Port	Gibt optional einen Port an. *Port* kann eine Portnummer oder der Name eines TCP-Dienstes mit einer Portnummer sein, die bei der Ausführung ermittelt wird. Wird kein Port angegeben, so wird der HTTP-Standardport 80 verwendet.
Pfad	Der Speicherort des vom angegebenen Server abzurufenden Archivs. Wird ein HTTP-Server verwendet, so muss das Package im Datastream-Format vorliegen.
opt_Schlüsselwörter	Legt die optionalen Schlüsselwörter für den Abruf eines Packages von einem HTTP-Server fest.

TABELLE 8–6 Optionale `package`-Schlüsselwörter für die Verwendung mit HTTP

Schlüsselwort	Definition des Werts
`timeout` *Min*	Mit dem Schlüsselwort `timeout` können Sie in Minuten angeben, wie lange maximal keine Daten vom HTTP-Server eingehen dürfen, bevor die Verbindung beendet, wieder hergestellt und an dem Punkt, an dem die Zeitüberschreitung aufgetreten ist, wieder aufgenommen wird. Setzen Sie `timeout` auf den Wert 0 (Null), wird die Verbindung bei Inaktivität nicht neu hergestellt. Sollte eine Verbindung nach einer Zeitüberschreitung neu aufgebaut werden, erfolgt ein neuer Versuch beim Anfang des Packages. Die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen.
`proxy` *Host:Port*	Das Schlüsselwort `proxy` ermöglicht es, einen Proxy-Host und einen Proxy-Port festzulegen. Mithilfe eines Proxy-Hosts können Sie Solaris-Packages jenseits einer Firewall abrufen. Wenn Sie das Schlüsselwort `proxy` verwenden, müssen Sie auch einen Proxy-Port angeben.

BEISPIEL 8–13 Hinzufügen eines Packages über HTTP

In diesem Beispiel werden mit dem Profilschlüsselwort `package` alle Packages hinzugefügt, die sich im Verzeichnis `Solaris 10` des HTTP-Speicherorts `http://package.central/Solaris_10` befinden. Sollten über einen Zeitraum von fünf Minuten keine Daten empfangen werden, wird erneut mit der Übertragung des Packages begonnen. Bereits empfangene Daten des betreffenden Packages werden in diesem Fall verworfen. Beide hier angegebenen Formate sind zulässig und möglich.

```
package SUNWnew add http package.central/Solaris_10 timeout 5

package SUNWnew add http://package.central/Solaris_10 timeout 5
```

BEISPIEL 8–14 Hinzufügen eines Packages über HTTP mit Proxy

In diesem Beispiel werden mit dem Profilschlüsselwort `package` alle Packages hinzugefügt, die sich im Verzeichnis `Solaris_10` des HTTP-Speicherorts `http://package.central/Solaris_10` befinden. Dabei wird angenommen, dass zwischen den Rechnern eine Firewall installiert ist. Um trotzdem eine Übertragung zu ermöglichen, wird mithilfe des `proxy`-Schlüsselworts ein Proxy angegeben.

```
package SUNWnew add http://package.central/Solaris_10 proxy webcache.east:8080
```

Lokales Gerät

Sie können Solaris-Packages auch von einem lokalen Gerät abrufen, wenn es sich dabei um ein dateisystemfähiges Gerät mit wahlfreiem Zugriff handelt, so zum Beispiel eine Diskette oder eine DVD. Verwenden Sie hierfür das Schlüsselwort `package` mit der folgenden Syntax:

`package` *Packagename* `add Lokales_Gerät` *Gerät Pfad Dateisystemtyp*

Packagename — Gibt den Namen des Packages an. Dieser muss im Format `SUNW`*Name* vorliegen. Detaillierte Informationen zu Packages und deren Namen können Sie auf einem bereits installierten System mit dem Befehl `pkginfo -l` anzeigen lassen.

Gerät — Gibt den Namen des Laufwerks an, auf dem das Solaris-Package gespeichert ist. Handelt es sich beim Gerätenamen um einen standardisierten Pfad, wird das Gerät direkt eingehängt. Wenn Sie einen Gerätenamen angeben, der kein standardisierter Pfad ist, so fügt das Installationsdienstprogramm den Abschnitt `/dev/dsk/` an den Pfad an.

Pfad — Gibt den Pfad zu dem gewünschten Solaris-Package an. Dieser wird relativ zum Root-Dateisystem (`/`) auf dem genannten Gerät angegeben.

Dateisystemtyp Gibt den Dateisystemtyp auf dem Gerät an. Wenn Sie keinen Dateisystemtyp angeben, versucht das Installationsdienstprogramm, ein UFS-Dateisystem einzuhängen. Sollte der UFS-Einhängevorgang fehlschlagen, wird versucht, ein HSFS-Dateisystem einzuhängen.

BEISPIEL 8–15 Hinzufügen eines Packages von einem lokalen Gerät mit UFS-Dateisystem

In diesem Beispiel wird mit dem Profilschlüsselwort `package` das Package `SUNWnew` aus dem Verzeichnis `/Solaris_10/Product` auf dem lokalen Gerät c0t6d0s0 hinzugefügt. Hierbei handelt es sich um ein UFS-Dateisystem.

```
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product ufs
```

BEISPIEL 8–16 Hinzufügen eines Packages von einem lokalen Gerät mit HSFS-Dateisystem

In diesem Beispiel wird mit dem Profilschlüsselwort `package` das Package `SUNWnew` aus dem Verzeichnis `/Solaris_10/Product` auf dem lokalen Gerät c0t6d0s0 hinzugefügt. Hierbei handelt es sich um ein HSFS-Dateisystem.

```
package SUNWnew add local_device c0t6d0s0 /Solaris_10/Product  hsfs
```

Lokale Datei

Sie können ein in dem Miniroot, von welchem Sie das System gebootet haben, gespeichertes Package als lokale Datei abrufen und installieren. Wenn Sie eine benutzerdefinierte JumpStart-Installation vornehmen, booten Sie das System von einer DVD, einer CD-ROM oder einer NFS-basierten Miniroot. Die Installationssoftware wird von dieser Miniroot geladen und ausgeführt. Folglich ist ein Package, das Sie auf der DVD, der CD-ROM oder in der NFS-basierten Miniroot gespeichert haben, als lokale Datei zugänglich. Verwenden Sie hierfür das Schlüsselwort `package` mit der folgenden Syntax:

`package` *Packagename* `add local_file` *Pfad*

Packagename Gibt den Namen des Packages an. Dieser muss im Format `SUNW`*Name* vorliegen. Detaillierte Informationen zu Packages und deren Namen können Sie auf einem bereits installierten System mit dem Befehl `pkginfo -l` anzeigen lassen.

Pfad Gibt den Speicherort des Packages an. Das System muss auf den Pfad wie auf eine lokale Datei zugreifen können, während das System von der Solaris Software-1-CD oder der Solaris Operating System-DVD aus gebootet wird. Auf `/net` kann das System nicht zugreifen, während es von der Solaris Software-1-CD oder der Solaris Operating System-DVD aus gebootet wird.

BEISPIEL 8–17 Hinzufügen eines Packages aus einer lokalen Datei

In diesem Beispiel wird mit dem Profilschlüsselwort `package` das Package `SUNWnew` aus dem Verzeichnis `/Solaris_10/Product` hinzugefügt.

```
package SUNWnew add local_file /Solaris_10/Product
```

Einschränkungen bei der Verwendung des Schlüsselworts `package`

Bitte achten Sie beim Einsatz des `package`-Schlüsselworts auf die folgenden Einschränkungen:

- Einige Packages sind erforderlich und können nicht gelöscht werden.
- Mit dem Profilschlüsselwort `package` können Sie Lokalisierungs-Packages nicht einzeln hinzufügen bzw. löschen. Verwenden Sie zum Hinzufügen von Lokalisierungs-Packages das Profilschlüsselwort `locale`.
- Es ist nicht möglich, Packages von einem FTP-Server oder einem lokalen Sicherungsmedium (z. B. Bandlaufwerk) abzurufen.
- Packages, die in der zu installierenden Solaris-Distribution enthalten sind, können nicht von anderen Speicherorten abgerufen werden. Wird ein Package aus der Solaris-Distribution angegeben, so darf folglich kein alternativer Speicherort folgen. Auf diese Weise bleibt später die Konsistenz mit dem Rest des installierten Systems gewahrt.
- Wenn eine ungeführte Installation ohne weitere Benutzereingriffe gewünscht ist, muss sich das Package mit dem Befehl `pkgadd` installieren lassen. Wenn gleichzeitig Softwaregruppen-Packages und Packages aus anderen Speicherorten installiert werden, muss dabei dieselbe `admin`-Datei verwendet werden.
  - Wenn HTTP als Abrufmethode verwendet wird, muss das Package im Stream-Format vorliegen.
  - Wenn das Package von einem NFS-Server, einem lokalen Gerät oder einer lokalen Datei abgerufen wird, sollte es im normalen Packaging-Format vorliegen. Dabei sollte der Verzeichnisname identisch mit dem Namen des zu installierenden Packages sein.
  - Wenn ein Package von einem externen Speicherort installiert werden soll und dieses Package ein anderes, gegenwärtig nicht installiertes Package voraussetzt, so wird es nicht installiert. In diesem Fall wird eine Fehlermeldung in das Installations-/Upgradeprotokoll geschrieben.
- Bei der Installation von Packages mithilfe eines Solaris Flash-Archivs befolgen Sie bitte folgende Richtlinien:
  - Alle installierten Packages müssen mit dem Archiv kompatibel sein.
  - Ist ein Package bereits im Archiv vorhanden, so überschreibt JumpStart das vorhandene Package.

Upgrades in Verbindung mit dem `package`-Schlüsselwort

Wenn Sie `package` für ein Upgrade verwenden, führt das JumpStart-Programm die folgenden Aktionen aus:

- Alle bereits auf dem System installierten Packages·werden automatisch aktualisiert.
- Wenn Sie *Packagename* `add` angeben und *Packagename* nicht auf dem System installiert ist, wird das Package installiert.
- Wenn Sie *Packagename* `delete` angeben und *Packagename* auf dem System installiert ist, wird das Package gelöscht, *bevor* das Upgrade beginnt.
- Wenn Sie *Packagename* `delete` angeben und *Packagename* nicht auf dem System installiert ist, wird das Package nicht installiert, wenn es Teil eines Clusters ist, das installiert werden soll.

`partitioning`-Profilschlüsselwort

`partitioning` *Typ*

`partitioning` definiert, wie die Festplatten während der Installation in Speicherbereiche für Dateisysteme aufgeteilt werden.

Wenn Sie `partitioning` im Profil nicht angeben, gilt standardmäßig der Partitionierungswert `default`.

Typ Verwenden Sie hier einen der folgenden Werte:

`default` Das JumpStart-Programm wählt die Festplatten aus und erstellt die Dateisysteme, in denen die angegebene Software installiert wird, mit Ausnahme aller Dateisysteme, die über das Schlüsselwort `filesys` angegeben werden. `rootdisk` wird zuerst ausgewählt. Das JumpStart-Programm verwendet zusätzliche Festplatten, wenn die angegebene Software nicht auf `rootdisk` passt.

`existing` Das JumpStart-Programm verwendet die vorhandenen Dateisysteme auf den Festplatten des Systems. Alle Dateisysteme mit Ausnahme von `/`, `/usr`, `/usr/openwin`, `/opt` und `/var` werden beibehalten. Das JumpStart-Programm verwendet das letzte Einhängepunktfeld aus dem Dateisystemsuperblock, um zu ermitteln, welchen Dateisystemeinhängepunkt der Speicherbereich darstellt.

Hinweis – Wenn Sie die Profilschlüsselwörter `filesys` und `partitioning existing` verwenden, müssen Sie für *Größe* den Wert `existing` angeben.

`explicit` Das JumpStart-Programm verwendet die Festplatten und erstellt die Dateisysteme, die über die Schlüsselwörter `filesys` angegeben werden. Wenn Sie nur das Root-Dateisystem (`/`) über das Schlüsselwort `filesys` angeben, wird die gesamte Solaris-Software im Root-Dateisystem (`/`) installiert.

Hinweis – Wenn Sie den Profilwert `explicit` verwenden, müssen Sie mit dem Schlüsselwort `filesys` die zu verwendenden Festplatten und die zu erstellenden Dateisysteme angeben.

`patch`-Profilschlüsselwort

`patch` *Patch-ID-Liste* | *patch_file Patch-Speicherort Optionale_Schlüsselwörter*]

Patch-ID-Liste Gibt die ID-Kennungen der zu installierenden Patches an. Hierbei muss es sich um eine kommagetrennte Liste von Solaris-Patch-IDs handeln. Die Patches werden in der Reihenfolge installiert, in der sie in der Liste auftauchen. Hinter den Kommata darf kein zusätzliches Leerzeichen stehen. Korrektes Beispiel: 112467-01,112765-02.

Patchdatei Eine Datei mit der Liste der Patches, die unter *Patch-Speicherort* gespeichert ist. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei aufgeführt sind.

Patch-Speicherort Gibt den Speicherort der Patches an. Folgende Arten von Speicherorten sind zulässig:

- NFS-Server
- HTTP-Server
- Local device
- Lokale Datei

opt_Schlüsselwörter Je nach Speicherort der Patches können verschiedene optionale Schlüsselwörter angegeben werden. Die folgenden Abschnitte beschreiben die möglichen Speicherorte und die dazugehörigen Schlüsselwörter.

NFS-Server

Wenn der fragliche Patch auf einem NFS-Server gespeichert ist, verwenden Sie für `patch` eine der folgenden Syntaxmöglichkeiten.

`patch` *Patch-ID-Liste* | *Patch-Datei* `nfs` *Servername:/Patch-Verzeichnis* `[retry` *n*`]`
`patch` *Patch-ID-Liste* | *Patch-Datei* `nfs://`*Servername/Patch-Verzeichnis* `[retry` *n*`]`

Patch-ID-Liste	Gibt die ID-Kennungen der zu installierenden Patches an. Hierbei muss es sich um eine kommagetrennte Liste von Solaris-Patch-IDs handeln. Die Patches werden in der Reihenfolge installiert, in der sie in der Liste auftauchen.
Patchdatei	Eine Datei mit der Liste der Patches, die unter *Patch-Speicherort* gespeichert ist. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei aufgeführt sind.
Server	Gibt den Namen des Servers an, auf dem die Patches gespeichert sind.
Patch-Verzeichnis	Gibt den Ort des Patch-Verzeichnisses auf dem angegebenen Server an. Die Patches müssen im Patch-Standardformat vorliegen.
retry n	Optionales Schlüsselwort. Das Installationsdienstprogramm versucht hiermit maximal *n* Mal, das Verzeichnis einzuhängen.

BEISPIEL 8–18 Hinzufügen von Patches aus einer geordneten Liste über NFS

In diesem Beispiel werden mit dem Profilschlüsselwort `patch` alle Patches hinzugefügt, die in der Datei `patch` im NFS-Patchverzeichnis `nfs://patch_master/Solaris/v10/patches` aufgeführt sind. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei `patch` vorkommen. Sollte das NFS-Einhängen fehlschlagen, werden maximal fünf Versuche unternommen.

```
patch patch_file nfs://patch_master/Solaris/v10/patches retry 5
```

BEISPIEL 8–19 Hinzufügen eines Patches über NFS

In diesem Beispiel fügt das Profilschlüsselwort `patch` die Patches 112467–01 und 112765–02 aus dem Patchverzeichnis `/Solaris/v10/patches` auf dem Server `patch_master` hinzu.

```
patch 112467-01,112765-02 nfs patch_master:/Solaris/v10/patches
```

HTTP-Server

Wenn der fragliche Patch auf einem HTTP-Server gespeichert ist, verwenden Sie für `patch` eine der folgenden Syntaxmöglichkeiten.

`patch` *Patch-ID-Liste* | *Patchdatei* `http://`*Servername [:port] Patch-Verzeichnis Optionale_HTTP-Schlüsselwörter*

`patch` *Patch-ID-Liste* | *Patchdatei* `http` *Servername [:port] Patch-Verzeichnis Optionale_HTTP-Schlüsselwörter*

Patch-ID-Liste	Gibt die ID-Kennungen der zu installierenden Patches an. Hierbei muss es sich um eine kommagetrennte Liste von Solaris-Patch-IDs handeln. Die Patches werden in der Reihenfolge installiert, in der sie in

	der Liste auftauchen. Hinter den Kommata darf kein zusätzliches Leerzeichen stehen. Korrektes Beispiel: 112467-01,112765-02.
Patchdatei	Eine Datei mit der Liste der Patches, die unter *Patch-Speicherort* gespeichert ist. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei aufgeführt sind.
Server	Gibt den Namen des Servers an, auf dem die Patches gespeichert sind.
Port	Gibt optional einen Port an. *Port* kann eine Portnummer oder der Name eines TCP-Dienstes mit einer Portnummer sein, die bei der Ausführung ermittelt wird. Wird kein Port angegeben, so wird der HTTP-Standardport 80 verwendet.
Patch-Verzeichnis	Der Speicherort der vom angegebenen Server abzurufenden Patches. Bei der Verwendung eines HTTP-Servers muss der Patch im JAR-Format vorliegen.
opt_Schlüsselwörter	Legt die optionalen Schlüsselwörter für den Abruf eines Patches von einem HTTP-Server fest.

TABELLE 8–7 Optionale `patch`-Schlüsselwörter für die Verwendung mit HTTP

Schlüsselwort	Definition des Werts
`timeout` *Min*	Mit dem Schlüsselwort `timeout` können Sie in Minuten angeben, wie lange maximal keine Daten vom HTTP-Server eingehen dürfen, bevor die Verbindung beendet, wieder hergestellt und an dem Punkt, an dem die Zeitüberschreitung aufgetreten ist, wieder aufgenommen wird. Setzen Sie `timeout` auf den Wert 0 (Null), wird die Verbindung bei Inaktivität nicht neu hergestellt. Sollte eine Verbindung nach einer Zeitüberschreitung neu aufgebaut werden, erfolgt ein neuer Versuch beim Anfang des Packages. Die vor der Zeitüberschreitung bereits abgerufenen Daten werden verworfen.
`proxy` *Host:Port*	Das Schlüsselwort `proxy` ermöglicht es, einen Proxy-Host und einen Proxy-Port festzulegen. Mithilfe eines Proxy-Hosts können Sie Solaris-Packages jenseits einer Firewall abrufen. Wenn Sie das Schlüsselwort `proxy` verwenden, müssen Sie auch einen Proxy-Port angeben.

BEISPIEL 8–20 Hinzufügen von Patches aus einer geordneten Liste über HTTP

In diesem Beispiel werden mit dem Profilschlüsselwort `patch` alle Patches hinzugefügt, die in der Datei `Patchdatei` im HTTP-Speicherort `http://patch.central/Solaris/v10/patches` aufgeführt sind. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei `Patchdatei` aufgeführt sind. Sollten über einen Zeitraum von fünf Minuten keine Daten empfangen werden, wird erneut mit der Übertragung des Patches begonnen. Bereits empfangene Daten des betreffenden Patches werden in diesem Fall verworfen.

BEISPIEL 8–20 Hinzufügen von Patches aus einer geordneten Liste über HTTP *(Fortsetzung)*

```
patch patch_file http://patch.central/Solaris/v10/patches timeout 5
```

BEISPIEL 8–21 Hinzufügen eines Patches über HTTP

In diesem Beispiel fügt das Profilschlüsselwort `patch` die Patches 112467–01 und 112765–02 vom Patchspeicherort `http://patch_master/Solaris/v10/patches` hinzu.

```
patch 112467-01,112765-02 http://patch.central/Solaris/v10/patches
```

Lokales Gerät

Sie können Solaris-Packages auch von einem lokalen Gerät abrufen, wenn es sich dabei um ein dateisystemfähiges Gerät mit wahlfreiem Zugriff handelt, so zum Beispiel eine Diskette oder eine DVD. Verwenden Sie hierfür das Schlüsselwort `patch` mit der folgenden Syntax:

`patch` *Patch-ID-Liste* | *Patchdatei* `local_device \`
Gerät Pfad Dateisystemtyp

Patch-ID-Liste Gibt die ID-Kennungen der zu installierenden Patches an. Hierbei muss es sich um eine kommagetrennte Liste von Solaris-Patch-IDs handeln. Die Patches werden in der Reihenfolge installiert, in der sie in der Liste auftauchen. Hinter den Kommata darf kein zusätzliches Leerzeichen stehen. Korrektes Beispiel: 112467-01,112765-02.

Patchdatei Eine Datei mit der Liste der Patches, die unter *Patch-Speicherort* gespeichert ist. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei aufgeführt sind.

Gerät Gibt den Namen des Laufwerks an, auf dem das Solaris-Package gespeichert ist. Handelt es sich beim Gerätenamen um einen standardisierten Pfad, wird das Gerät direkt eingehängt. Wenn Sie einen Gerätenamen angeben, der kein standardisierter Pfad ist, so fügt das Installationsdienstprogramm den Abschnitt `/dev/dsk/` an den Pfad an.

Pfad Gibt den Pfad zum Solaris-Patch an, relativ zum Root-Dateisystem (`/`) auf dem von Ihnen angegebenen Gerät.

Dateisystemtyp Gibt den Dateisystemtyp auf dem Gerät an. Wenn Sie keinen Dateisystemtyp angeben, versucht das Installationsdienstprogramm, ein UFS-Dateisystem einzuhängen. Sollte der UFS-Einhängevorgang fehlschlagen, wird versucht, ein HSFS-Dateisystem einzuhängen.

BEISPIEL 8–22 Hinzufügen von Patches aus einer geordneten Liste von einem lokalen Gerät

In diesem Beispiel fügt das Profilschlüsselwort `patch` alle Patches, die in der Datei `patch_file` des Verzeichnisses `/Solaris_10/patches` des Gerätes c0t6d0s0 enthalten sind, hinzu. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei `Patchdatei` vorkommen.

```
patch patch_file c0t6d0s0 /Solaris_10/patches
```

BEISPIEL 8–23 Hinzufügen eines Patches von einem lokalen Gerät

In diesem Beispiel fügt das Profilschlüsselwort `patch` die Patches 112467–01 und 112765–02 aus dem Patchverzeichnis `/Solaris_10/patches` vom lokalen Gerät c0t6d0s0 hinzu.

```
patch 112467-01,112765-02 local_device c0t6d0s0 /Solaris_10/patches
```

Lokale Datei

Sie können einen in dem Miniroot, von welchem Sie das System gebootet haben, gespeicherten Patch als lokale Datei abrufen und installieren. Wenn Sie eine benutzerdefinierte JumpStart-Installation vornehmen, booten Sie das System von einer DVD, einer CD-ROM oder einer NFS-basierten Miniroot. Die Installationssoftware wird von dieser Miniroot geladen und ausgeführt. Folglich ist ein Patch, den Sie auf der DVD, der CD-ROM oder in der NFS-basierten Miniroot gespeichert haben, als lokale Datei zugänglich. Verwenden Sie hierfür das Schlüsselwort `patch` mit der folgenden Syntax:

`patch` *Patch-ID-Liste* | *Patchdatei* `local_file` *Patch-Verzeichnis*

Patch-ID-Liste — Gibt die ID-Kennungen der zu installierenden Patches an. Hierbei muss es sich um eine kommagetrennte Liste von Solaris-Patch-IDs handeln. Die Patches werden in der Reihenfolge installiert, in der sie in der Liste auftauchen. Hinter den Kommata darf kein zusätzliches Leerzeichen stehen. Korrektes Beispiel: 112467-01,112765-02.

Patchdatei — Eine Datei mit der Liste der Patches, die unter *Patch-Speicherort* gespeichert ist. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei aufgeführt sind.

Patch-Verzeichnis — Gibt den Speicherort für das Patch-Verzeichnis an. Das System muss auf das Patch-Verzeichnis wie auf eine lokale Datei zugreifen können, während das System von der Solaris Software-1-CD oder der Solaris Operating System-DVD aus gebootet wird. Auf `/net` kann das System nicht zugreifen, während es von der SolarisInstallationCD1 oder der Solaris Operating System-DVD aus gebootet wird.

BEISPIEL 8–24 Hinzufügen von Patches aus einer geordneten Liste von einer lokalen Datei

In diesem Beispiel werden mit dem Profilschlüsselwort `patch` alle Patches hinzugefügt, die in der Datei `Patchdatei` im Verzeichnis `/Solaris_10/patches` aufgeführt sind. Die Patches werden in der Reihenfolge installiert, in der sie in der Datei `Patchdatei` vorkommen.

```
patch patch_cal_file /Solaris_10/patches
```

BEISPIEL 8–25 Hinzufügen eines Patches aus einer lokalen Datei

In diesem Beispiel fügt das Profilschlüsselwort `patch` die Patches 112467–01 und 112765–02 aus dem Patchverzeichnis `/Solaris_10/patches` hinzu.

```
patch 112467-01,112765-02 local_file /Solaris_10/patches
```

Einschränkungen bei der Verwendung des Schlüsselworts `patch`

Bitte achten Sie beim Einsatz des `patch`-Schlüsselworts auf die folgenden Einschränkungen:

- Es ist nicht möglich, Patches von einem FTP-Server oder einem lokalen Sicherungsmedium (z. B. Bandlaufwerk) abzurufen.
- Es ist nicht möglich, signierte Patches hinzuzufügen.
- Die Patches müssen sich mit dem Befehl `patchadd` installieren lassen.
- Patches, die wiederum einen gegenwärtig nicht installierten Patch voraussetzen, werden nicht installiert. In diesem Fall wird eine Fehlermeldung in das Installations-/Upgradeprotokoll geschrieben.
- Die für eine korrekte Installation der Patches erforderliche Reihenfolge müssen Sie selbst ermitteln.

`pool`-Profilschlüsselwort (nur ZFS)

Das Schlüsselwort `pool` definiert die Installation eines ZFS-Root-Pools. Das Pool wird mit einer durch das Schlüsselwort `cluster` angegebenen Softwaregruppe installiert. Zum Erstellen eines neuen Root-Pools werden die Optionen *Pool-Größe*, *Swap-Größe*, *Speicherabzugsgröße* und *Virtuelle_Geräteliste* benötigt.

Eine vollständige Beschreibung des Schlüsselworts `pool` und anderer Schlüsselwörter, die für ZFS-Root-Pools verwendet werden können, finden Sie unter „ `pool`-Profilschlüsselwort (nur ZFS)“ auf Seite 179.

root_device-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort root_device kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendung dieses Schlüsselworts ist auf ein Einzelsystem für ZFS-Installationen beschränkt.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Eine Beschreibung zur Verwendung des Schlüsselworts root_device für die Installation eines ZFS-Root-Pools finden Sie in „JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick)“ auf Seite 177.

root_device *Speicherbereich*

root_device legt die Root-Festplatte des Systems fest. „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165 enthält weitere Informationen hierzu.

Hinweis – Die Root-Festplatte wird vom JumpStart-Programm ermittelt und legt fest, wo das Betriebssystem installiert wird. Die rules-Datei verwendet das Probe-Schlüsselwort ?rootdisk“, aber dieses Schlüsselwort wird anders als das Schlüsselwort "rootdisk" im JumpStart-Profil verwendet. Sie können den Installationsort nicht mithilfe des Probe-Schlüsselworts rootdisk in der rules-Datei festlegen. Das Probe-Schlüsselwort rootdisk legt fest, von wo während der Installation gebootet wird. Siehe Tabelle 8–10.

Bei einem Upgrade eines Systems legt root_device das Root-Dateisystem (/) und die über dessen /etc/vfstab-Datei eingehängten Dateisysteme fest, die aktualisiert werden sollen. Sie müssen root_device angeben, wenn mehr als ein Root-Dateisystem (/) auf einem System aktualisiert werden kann. Sie müssen *Speicherbereich* im Format c*w* t*xdy* s*z* oder c*xdy* s*z* angeben.

Beachten Sie Folgendes, wenn Sie das Schlüsselwort root_device verwenden:

- Wenn Sie root_device auf einem System mit nur einer Festplatte angeben, müssen root_device und die Festplatte übereinstimmen. Außerdem müssen alle filesys-Schlüsselwörter, die das Root-Dateisystem (/) angeben, mit root_device übereinstimmen.
- Beim Upgrade eines RAID-1-Volumes (Mirrors) sollte es sich bei dem für root_device angegebenen Wert um eine Seite des RAID-1-Volumes (Mirrors) handeln. Die andere Seite der gespiegelten Platte wird automatisch aktualisiert.

BEISPIEL 8–26 `root_device`-Profilschlüsselwort

```
root_device c0t0d0s2
```

So wird die Root-Festplatte des Systems festgelegt

Die Root-Festplatte eines Systems ist die Festplatte, die das Root-Dateisystem (`/`) enthält. In einem Profil können Sie anstelle eines Festplattennamens die Variable `rootdisk` verwenden, die vom JumpStart-Programm auf die Root-Festplatte des Systems gesetzt wird. Tabelle 8–8 beschreibt, wie das JumpStart-Programm die Root-Festplatte für die Installation festlegt.

Hinweis – Das JumpStart-Programm kann die Größe einer Root-Festplatte nur bei einer Neuinstallation festlegen. Bei einem Upgrade kann die Root-Festplatte eines Systems nicht modifiziert werden.

TABELLE 8–8 So legt JumpStart die Root-Festplatte eines Systems fest (Neuinstallation)

Abschnitt	Aktion
1	Wenn das Schlüsselwort `root_device` im Profil angegeben wird, setzt das JumpStart-Programm `rootdisk` auf das Root-Gerät.
2	Wenn `rootdisk` nicht gesetzt ist und das Schlüsselwort `boot_device` im Profil angegeben wird, setzt das JumpStart-Programm `rootdisk` auf das Boot-Gerät.
3	Wenn `rootdisk` nicht gesetzt ist und ein Eintrag des Formats `filesys` c*w*t*x*d*y*s*z* *Größe* `/` im Profil angegeben wird, setzt das JumpStart-Programm `rootdisk` auf die in diesem Eintrag angegebene Festplatte.
4	Wenn `rootdisk` nicht gesetzt ist und ein Eintrag `rootdisk.s`*n* im Profil angegeben wird, durchsucht das JumpStart-Programm die Festplatten des Systems in der Kernel-Sondierungsreihenfolge nach einem vorhandenen Root-Dateisystem auf dem angegebenen Speicherbereich. Wenn eine Festplatte gefunden wird, setzt das JumpStart-Programm `rootdisk` auf die entsprechende Festplatte.
5	Wenn `rootdisk` nicht gesetzt ist und im Profil `partitioning existing` angegeben wird, durchsucht das JumpStart-Programm die Festplatten des Systems in der Kernel-Sondierungsreihenfolge nach einem vorhandenen Root-Dateisystem. Wenn kein oder mehr als ein Root-Dateisystem gefunden wird, tritt ein Fehler auf. Wenn ein Root-Dateisystem gefunden wird, setzt das JumpStart-Programm `rootdisk` auf die entsprechende Festplatte.
6	Wenn `rootdisk` nicht gesetzt ist, setzt das JumpStart-Programm `rootdisk` auf die Festplatte, auf der das Root-Dateisystem (`/`) installiert wird.

`system_type`-Profilschlüsselwort

`system_type` *Typ-Schalter*

`system_type` definiert den Typ des Systems, auf dem Solaris installiert werden soll.

Typ-Schalter steht für die Option `standalone` oder `server`. Hiermit können Sie den Typ des Systems angeben, auf dem die Solaris-Software installiert werden soll. Wenn Sie `system_type` in einem Profil nicht angeben, wird standardmäßig `standalone` verwendet.

`usedisk`-Profilschlüsselwort (UFS und ZFS)

Hinweis – Das Schlüsselwort `usedisk` kann verwendet werden, wenn Sie ein UFS-Dateisystem oder ein ZFS-Root-Pool installieren. Die Verwendungsweise für dieses Schlüsselwort ist in UFS- und ZFS-Installationen gleich.

- Eine vollständige Liste der Schlüsselwörter, die in UFS- bzw. ZFS-Installationen verwendet werden können, finden Sie in Tabelle 8–2.
- Informationen zum Ausführen einer ZFS-Installation finden Sie in Kapitel 9, „Installation eines ZFS-Root-Pools mit JumpStart“.

`usedisk` *Festplattenname* `...`

Standardmäßig verwendet das JumpStart-Programm alle betriebsbereiten Festplatten im System, wenn Sie `partitioning default` angeben. Mit dem Schlüsselwort `usedisk` können Sie eine oder mehr Festplatten, die vom JumpStart-Programm verwendet werden sollen, explizit angeben. Sie müssen *Festplattenname* im Format c*x*t *y*d*z* oder c *y*d*z* angeben, zum Beispiel `c0t0d0` oder `c0d0s0`.

Wenn Sie `usedisk` in einem Profil angeben, verwendet das JumpStart-Programm nur die Festplatten, die Sie nach dem Schlüsselwort `usedisk` angeben.

Hinweis – Sie können die Schlüsselwörter `usedisk` und `dontuse` nicht in demselben Profil verwenden.

Benutzerdefinierte JumpStart-Umgebungsvariablen

In Begin- und Finish-Skripten können Sie benutzerdefinierte JumpStart-Umgebungsvariablen verwenden. Sie können zum Beispiel ein Begin-Skript schreiben, das die Festplattengröße, `SI_DISKSIZES`, extrahiert, und je nach der vom Skript extrahierten Festplattengröße festlegen, ob bestimmte Packages auf einem System installiert werden sollen.

In Umgebungsvariablen werden zu einem System erfasste Informationen gespeichert. Die Variablen werden im Allgemeinen je nach den in der Datei `rules` verwendeten Rule-Schlüsselwörtern und -Werten gesetzt.

Informationen darüber, welches Betriebssystem bereits auf einem System installiert ist, steht in `SI_INSTALLED` erst zur Verfügung, nachdem das Schlüsselwort `installed` verwendet wurde.

Tabelle 8–9 beschreibt diese Variablen und deren Werte.

TABELLE 8–9 Umgebungsvariablen für die Installation

Umgebungsvariable	Wert
SI_ARCH	Die Hardwarearchitektur des Installationsclients. Die Variable SI_ARCH wird gesetzt, wenn Sie das Schlüsselwort arch in der Datei rules verwenden.
SI_BEGIN	Der Name des Begin-Skripts, sofern eins verwendet wird.
SI_CLASS	Der Name des Profils, das zur Installation des Installationsclients verwendet wird.
SI_DISKLIST	Eine Liste mit den durch Kommas getrennten Festplattennamen des Installationsclients. Die Variable SI_DISKLIST wird gesetzt, wenn das Schlüsselwort disksize verwendet und in der Datei rules eine Entsprechung gefunden wird. Die Variablen SI_DISKLIST und SI_NUMDISKS dienen zum Festlegen der physischen Festplatte, die für rootdisk verwendet werden soll. rootdisk ist in „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165 beschrieben.
SI_DISKSIZES	Eine Liste mit den durch Kommas getrennten Festplattengrößen des Installationsclients. Die Variable SI_DISKSIZES wird gesetzt, wenn das Schlüsselwort disksize verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_DOMAINNAME	Der Domänenname. Die Variable SI_DOMAINNAME wird gesetzt, wenn das Schlüsselwort domainname verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_FINISH	Der Name des Finish-Skripts, sofern eins verwendet wird.
SI_HOSTADDRESS	Die IP-Adresse des Installationsclients.
SI_HOSTNAME	Der Host-Name des Installationsclients. Die Variable SI_HOSTNAME wird gesetzt, wenn das Schlüsselwort hostname verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_INSTALLED	Der Gerätename einer Festplatte, auf der ein spezifisches Betriebssystem wie z.&;B. Solaris, SunOS oder System V installiert ist. Die Variable SI_INSTALLED wird gesetzt, wenn das Schlüsselwort installed verwendet wird und in der Datei rules einen entsprechenden Wert besitzt. SI_INST_OS und SI_INST_VER dienen dazu, den Wert von SI_INSTALLED festzulegen.
SI_INST_OS	Der Name des Betriebssystems. SI_INST_OS und SI_INST_VER dienen dazu, den Wert von SI_INSTALLED festzulegen.
SI_INST_VER	Die Version des Betriebssystems. SI_INST_OS und SI_INST_VER dienen dazu, den Wert von SI_INSTALLED festzulegen.
SI_KARCH	Die Kernel-Architektur des Installationsclients. Die Variable SI_KARCH wird gesetzt, wenn das Schlüsselwort karch verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_MEMSIZE	Die Größe des physischen Hauptspeichers auf dem Installationsclient. Die Variable SI_MEMSIZE wird gesetzt, wenn das Schlüsselwort memsize verwendet und in der Datei rules eine Entsprechung gefunden wird.

TABELLE 8–9 Umgebungsvariablen für die Installation *(Fortsetzung)*

Umgebungsvariable	Wert
SI_MODEL	Die Modellbezeichnung des Installationsclients. Die Variable SI_MODEL wird gesetzt, wenn das Schlüsselwort model verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_NETWORK	Die Netzwerknummer des Installationsclients. Die Variable SI_NETWORK wird gesetzt, wenn das Schlüsselwort network verwendet und in der Datei rules eine Entsprechung gefunden wird.
SI_NUMDISKS	Die Anzahl der Festplatten eines Installationsclients. Die Variable SI_NUMDISKS wird gesetzt, wenn das Schlüsselwort disksize verwendet und in der Datei rules eine Entsprechung gefunden wird. Die Variablen SI_NUMDISKS und SI_DISKLIST dienen zum Festlegen der physischen Festplatte, die für rootdisk verwendet werden soll. rootdisk ist in „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165 beschrieben.
SI_OSNAME	Das Betriebssystem-Release im Solaris-Softwareabbild. Sie können die Variable SI_OSNAME zum Beispiel in einem Skript verwenden, wenn Sie die Solaris-Software auf Systemen installieren, die auf der Version des Betriebssystems im Abbild der Solaris Operating System-DVD bzw. der Solaris Software-1-CD basieren.
SI_ROOTDISK	Der Gerätename der Festplatte, die durch den logischen Namen rootdisk dargestellt wird. Die Variable SI_ROOTDISK wird gesetzt, wenn das Schlüsselwort disksize oder installed in der Datei rules auf rootdisk gesetzt ist. Die Variable SI_ROOTDISK legt fest, von welchem Gerät während der Installation gebootet wird. **Hinweis –** Sie können den Installationsort nicht mithilfe des Probe-Schlüsselworts rootdisk in der rules-Datei festlegen. Weitere Informationen zur Variablen rootdisk, die in einem JumpStart-Profil festgelegt wird, finden Sie unter „So wird die Root-Festplatte des Systems festgelegt“ auf Seite 165.
SI_ROOTDISKSIZE	Die Größe der Festplatte, die durch den logischen Namen rootdisk dargestellt wird. Die Variable SI_ROOTDISKSIZE wird gesetzt, wenn das Schlüsselwort disksize oder installed in der Datei rules auf rootdisk gesetzt ist.
SI_TOTALDISK	Die Gesamtgröße des physischen Festplattenspeichers auf dem Installationsclient. Die Variable SI_TOTALDISK wird gesetzt, wenn das Schlüsselwort totaldisk verwendet und in der Datei rules eine Entsprechung gefunden wird.

Probe-Schlüsselwörter und -Werte

Tabelle 8–10 beschreibt jedes Rule-Schlüsselwort und das entsprechende Probe-Schlüsselwort.

Hinweis – Stellen Sie Probe-Schlüsselwörter immer möglichst an den Anfang der Datei rules.

TABELLE 8–10 Beschreibung von Probe-Schlüsselwörtern

Rule-Schlüsselwort	Entsprechendes Probe-Schlüsselwort	Beschreibung des Probe-Schlüsselworts
`any`	Keinen	
`arch`	`arch`	Ermittelt die Kernel-Architektur, i386 oder SPARC, und setzt `SI_ARCH`.
`disksize`	`disks`	Gibt die Größe der Festplatten eines Systems in MB in der Kernel-Sondierungsreihenfolge, c0t3d0s0, c0t3d0s1, c0t4d0s0, zurück. `disksize` setzt `SI_DISKLIST`, `SI_DISKSIZES`, `SI_NUMDISKS` und `SI_TOTALDISK`.
`domainname`	`domainname`	Gibt den NIS- oder NIS+-Domänennamen eines Systems oder nichts zurück und setzt `SI_DOMAINNAME`. Das Schlüsselwort `domainname` gibt die Ausgabe von `domainname(1M)` zurück.
`hostaddress`	`hostaddress`	Gibt die IP-Adresse eines Systems zurück, also die erste Adresse, die in der Ausgabe von `ifconfig(1M)` `-a` aufgeführt ist und nicht lo0 lautet, und setzt `SI_HOSTADDRESS`.
`hostname`	`hostname`	Gibt den Host-Namen eines Systems zurück, also die Ausgabe von `uname(1)` `-n`, und setzt `SI_HOSTNAME`.
`installed`	`installed`	Gibt die Versionsbezeichnung der aktuellen Solaris-Installation des Systems zurück und setzt `SI_ROOTDISK` und `SI_INSTALLED`. Wenn das JumpStart-Programm ein Solaris-Release findet, aber die Version nicht ermitteln kann, wird die Version `SystemV` zurückgegeben.
`karch`	`karch`	Gibt die Plattformgruppe eines Systems zurück, zum Beispiel i86pc oder sun4u, und setzt `SI_KARCH`. Eine Liste der Plattformnamen finden Sie im *Solaris Handbuch zur Hardware-Plattform von Sun* unter·`http://docs.sun.com`.
`memsize`	`memsize`	Gibt die Größe des physischen Hauptspeichers eines Systems in MB zurück und setzt `SI_MEMSIZE`.
`model`	`model`	Gibt den Plattformnamen eines Systems zurück und setzt `SI_MODEL`. Eine Liste der Plattformnamen finden Sie im *Solaris Handbuch zur Hardware-Plattform von Sun* unter·`http://docs.sun.com`.
`network`	`network`	Gibt die Netzwerknummer eines System zurück, die das JumpStart-Programm mithilfe eines logischen UND zwischen der IP-Adresse und der Teilnetzmaske des Systems ermittelt. Die IP-Adresse und die Teilnetzmaske des Systems werden aus der ersten in der Ausgabe des Befehls `ifconfig(1M)` `-a` aufgeführten Adresse extrahiert, die nicht lo0 lautet. Das Schlüsselwort `network` setzt `SI_NETWORK`.
`osname`	`osname`	Gibt die Version und den Betriebssystemnamen des Betriebssystems Solaris zurück, das auf einer CD gefunden wurde, und setzt `SI_OSNAME`. Wenn das JumpStart-Programm ein Solaris-Release findet, aber die Version nicht ermitteln kann, wird die Version `SystemV` zurückgegeben.
	`rootdisk`	Gibt den Namen und die Größe in MB der Root-Festplatte eines Systems zurück und setzt `SI_ROOTDISK`.

TABELLE 8–10 Beschreibung von Probe-Schlüsselwörtern *(Fortsetzung)*

Rule-Schlüsselwort	Entsprechendes Probe-Schlüsselwort	Beschreibung des Probe-Schlüsselworts
`totaldisk`	`totaldisk`	Gibt die Gesamtgröße des Festplattenspeichers eines Systems (in MB) zurück und setzt `SI_TOTALDISK`. Diese Angabe umfasst alle betriebsbereiten Festplatten, die an das System angeschlossen sind.

◆ ◆ ◆ KAPITEL 9

Installation eines ZFS-Root-Pools mit JumpStart

Dieses Kapitel enthält alle Informationen, die zum Ausführen einer JumpStart-Installation für ZFS-Root-Pools erforderlich sind. Die folgenden Abschnitte enthalten Planungsinformationen, Profilbeispiele sowie Beschreibungen von Profilschlüsselwörtern.

- „JumpStart-Installation für ein ZFS-Root-Dateisystem (/) (Übersicht und Planung)“ auf Seite 171
- „JumpStart-Profilbeispiele für ein ZFS-Root-Pool“ auf Seite 173
- „JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick)“ auf Seite 177

JumpStart-Installation für ein ZFS-Root-Dateisystem (/) (Übersicht und Planung)

Dieses Kapitel enthält die Informationen, die Sie zum Erstellen eines JumpStart-Profils zur Installation eines ZFS-Root-Pools benötigen.

Hinweis – Wenn Sie ein UFS-Root-Dateisystem (/) erstellen möchten, werden alle existierenden Profilschlüsselwörter wie in vorherigen Solaris-Releases verwendet. Eine Liste mit UFS-Profilschlüsselwörtern finden Sie in Kapitel 8, „Benutzerdefinierte JumpStart-Installation (Referenz)“.

Ein ZFS-spezifisches Profil muss das Schlüsselwort `pool` enthalten. Das Schlüsselwort `pool` installiert ein neues Root-Pool; dabei wird standardmäßig eine neue Boot-Umgebung erstellt. Sie können den Namen der Boot-Umgebung eingeben und einen eigenen `/var`-Dataset mit den vorhandenen Schlüsselwörtern `bootenv installbe` und den neuen `bename`- und `dataset` -Optionen erstellen. Einige in UFS-spezifischen Profilen gültige Schlüsselwörter sind in einem ZFS-spezifischen Profil nicht zulässig, so z.&;B. Schlüsselwörter zum Erstellen von UFS-Einhängepunkten.

Umfassende Informationen zur ZFS-Planung finden Sie in Kapitel 6, „Installation eines ZFS-Root-Dateisystems (Planung)“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

Einschränkungen für eine JumpStart-Installation für ein ZFS-Root-Pool

Berücksichtigen Sie folgende Aspekte, bevor Sie sich für eine JumpStart-Installation eines bootfähigen ZFS-Root-Pools entscheiden.

TABELLE 9–1 JumpStart-Einschränkungen für ZFS-Root-Pools

Einschränkung	Beschreibung	Weitere Informationen

Bei einer JumpStart-Installation können Sie zum Erstellen eines bootfähigen ZFS-Root-Pools kein vorhandenes ZFS-Speicherpool verwenden.	Sie müssen ein neues ZFS-Speicherpool mit einer Syntax wie der folgenden erstellen:
	`pool rpool 20G 4G 4G c0t0d0s0`
	Die gesamte Zeile für das Schlüsselwort `pool` ist erforderlich, da Sie kein vorhandenes Pool verwenden können. Das Schlüsselwort `bootenv` ist optional. Wenn Sie das Schlüsselwort `bootenv` nicht verwenden, wird eine Standard-Bootumgebung erstellt. Beispiel:
	`install_type initial_install`
	`cluster SUNWCall`
	`pool rpool 20G 4g 4g any`
	`bootenv installbe bename newBE`
		„`pool`-Profilschlüsselwort (nur ZFS)“ auf Seite 179

Pools können nicht auf gesamten Festplatten,	sondern nur auf Festplattenbereichen erstellt werden. Wenn Sie im Profil ein Pool auf einer gesamten Festplatte erstellen (z.&;B. `c0t0d0`) schlägt die Installation fehl. Es wird dann in etwa die folgende Fehlermeldung ausgegeben. `Invalid disk name (c0t0d0)`	
Einige in UFS-spezifischen Profilen gültige Schlüsselwörter sind in einem ZFS-spezifischen Profil nicht zulässig, so z.&;B. Schlüsselwörter zum Erstellen von UFS-Einhängepunkten.		„Profilschlüsselwörter im Überblick“ auf Seite 116

TABELLE 9–1 JumpStart-Einschränkungen für ZFS-Root-Pools *(Fortsetzung)*

Einschränkung	Beschreibung	Weitere Informationen
Sie können kein Upgrade mit JumpStart durchführen, sondern müssen dafür Solaris Live Upgrade verwenden.	Mit Solaris Live Upgrade können Sie eine Kopie des aktuell ausgeführten·Systems erstellen. Für diese Kopie kann dann das Upgrade durchgeführt werden. Danach wird sie aktiviert, wodurch sie zum aktuell ausgeführten·System wird.	Kapitel 11, „Solaris Live Upgrade und ZFS (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

JumpStart-Profilbeispiele für ein ZFS-Root-Pool

Dieser Abschnitt enthält Beispiele für ZFS-spezifische JumpStart-Profile.

Hinweis – Damit ein ZFS-Root-Pool upgrade- und bootfähig ist, darf es nicht auf einer gesamten Festplatte, sondern muss auf Festplattenbereichen erstellt werden. Wenn Sie in einem Profil ein Pool auf einer gesamten Festplatte wie z.&;B. `c0t0d0` erstellen, wird eine Fehlermeldung wie die folgende ausgegeben.

```
Invalid disk name (c0t0d0)
```

BEISPIEL 9–1 Installation eines gespiegelten ZFS-Root-Pools

```
install_type initial_install
cluster SUNWCall
pool newpool auto auto auto mirror c0t0d0s0 c0t1d0s0
bootenv installbe bename solaris10_6
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type initial_install` — Das Schlüsselwort `install_type` ist in jedem Profil erforderlich. Das Schlüsselwort `initial_install` führt eine Erstinstallation aus, bei der in einem neuen ZFS-Root-Pool ein neues Solaris-Betriebssystem installiert wird.

`cluster` — Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert. Weitere Informationen zu Softwaregruppen finden Sie unter „Empfohlener Festplattenspeicher für Softwaregruppen“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

BEISPIEL 9–1 Installation eines gespiegelten ZFS-Root-Pools *(Fortsetzung)*

`pool` — Das Schlüsselwort `pool` legt die Eigenschaften des neuen ZFS-Root-Pools fest.

- *Neues_Pool* — Legt den Namen des Root-Pools fest.
- `auto` — Legt die Größe der Festplattenpartitionen automatisch fest. Die Größe hängt von der Kapazität der angegebenen Festplattenpartitionen ab.
- `auto` — Die Größe des Swap-Bereichs wird mit dem Schlüsselwort `auto` automatisch festgelegt. Die Standardkapazität beträgt die Hälfte der Größe des physischen Speichers, jedoch mindestens 512&;MB und maximal 2&;GB. Mithilfe der Option `size` können Sie eine Kapazität außerhalb dieses Bereichs festlegen.
- `auto` — Die Kapazität des Speicherabzugsgeräts wird automatisch eingestellt.
- `mirror` — Die Datenspiegelungskonfiguration von Festplatten wird durch das Schlüsselwort `mirror` sowie Festplattenbereiche mit den Werten *c0t0d0s0* und *c0t1d0s0* festgelegt.

`bootenv` — `installbe` ändert die Eigenschaften der während der Installation erstellten Standard-Bootumgebung.

- `bename` — Benennt die neue Bootumgebung als *solaris10_6*.

BEISPIEL 9–2 Anpassen der Festplattenkapazität für ein ZFS-Root-Pool

```
install_type initial_install
cluster SUNWCall
pool newpool 80g 2g 2g mirror any any
bootenv installbe bename solaris10_6
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type initial_install` — Das Schlüsselwort `install_type` ist in jedem Profil erforderlich. Das Schlüsselwort `initial_install` führt

BEISPIEL 9–2 Anpassen der Festplattenkapazität für ein ZFS-Root-Pool *(Fortsetzung)*

	eine Erstinstallation aus, bei der in einem neuen ZFS-Root-Pool ein neues Solaris-Betriebssystem installiert wird.
`cluster`	Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert. Weitere Informationen zu Softwaregruppen finden Sie unter „Empfohlener Festplattenspeicher für Softwaregruppen“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.
`pool`	Das Schlüsselwort `pool` legt die Eigenschaften des neuen ZFS-Root-Pools fest. *Neues_Pool* — Legt den Namen des Root-Pools fest. `80g` — Legt die Kapazität des Plattenbereiches fest. `2g` — Die Kapazität des Swap- und des Speicherabzugbereichs beträgt jeweils 2 GB. `mirror` — Die Datenspiegelungskonfiguration von Festplatten wird durch das Schlüsselwort `mirror` sowie Festplattenbereiche mit den Werten *c0t0d0s0* und *c0t1d0s0* festgelegt. Die Option `any` in der Datenspiegelungskonfiguration sucht zwei verfügbare Datenträger, deren Kapazität zum Erstellen eines 80-GB-Pools ausreicht. Die Installation schlägt fehl, falls zwei solche Datenträger nicht verfügbar sind.
`bootenv`	`installbe` ändert die Eigenschaften der während der Installation erstellten Standard-Bootumgebung. `bename` — Benennt die neue Bootumgebung als *solaris10_6*.

BEISPIEL 9–3 Festlegen des Betriebssystem-Installationsorts

```
install_type initial_install
cluster SUNWCall
root_device c0t0d0s0
pool nrpool auto auto auto rootdisk.s0
bootenv installbe bename bnv dataset /var
```

In der folgenden Liste sind einige Schlüsselwörter und Werte aus diesem Beispiel beschrieben.

`install_type initial_install`
: Das Schlüsselwort `install_type` ist in jedem Profil erforderlich. Das Schlüsselwort `initial_install` führt eine Erstinstallation aus, bei der in einem neuen ZFS-Root-Pool ein neues Solaris-Betriebssystem installiert wird.

`cluster`
: Die gesamte Solaris-Softwaregruppe, `SUNWCall`, wird auf dem System installiert. Weitere Informationen zu Softwaregruppen finden Sie unter „Empfohlener Festplattenspeicher für Softwaregruppen“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades*.

`root_device`
: Legt den Plattenbereich fest, auf dem das Betriebssystem installiert werden soll. *c0t0d0s0* definiert die spezifische Festplatte und den Plattenbereich für das Betriebssystem.

`pool`
: Das Schlüsselwort `pool` legt die Eigenschaften des neuen ZFS-Root-Pools fest.

  nrpool
  : Legt den Namen des Root-Pools fest.

  `auto`
  : Legt die Größe der Festplattenpartitionen automatisch fest. Die Größe hängt von der Kapazität der angegebenen Festplattenpartitionen ab.

  `auto`
  : Die Größe des Swap-Bereichs wird mit dem Schlüsselwort `auto` automatisch festgelegt. Die Standardkapazität beträgt die Hälfte der Größe des physischen Speichers, jedoch mindestens 512&;MB und maximal 2&;GB. Mithilfe der Option `size` können Sie eine Kapazität außerhalb dieses Bereichs festlegen.

BEISPIEL 9–3 Festlegen des Betriebssystem-Installationsorts *(Fortsetzung)*

	`auto`	Die Kapazität des Speicherabzugsgeräts wird automatisch eingestellt.
	`rootdisk.s0`	Das Gerät, mit dem das Root-Pool erstellt wurde, wird als Bereich&;0 angegeben.
`bootenv`	`installbe` ändert die Eigenschaften der während der Installation erstellten Standard-Bootumgebung.	
	`bename`	Benennt die neue Bootumgebung als *bnv*.
	`dataset`	Erstellt ein vom ROOT-Dataset getrennten, eigenen `/var`-Dataset. `/var` ist der einzige Wert für `dataset` .

JumpStart-Schlüsselwörter für ein ZFS-Root-Dateisystem (/) (Überblick)

Dieser Abschnitt enthält Beschreibungen einiger ZFS-spezifischer Schlüsselwörter, die in einem JumpStart-Profil verwendet werden können. Die Verwendungsweise der in diesem Abschnitt beschriebenen Schlüsselwörter unterscheidet sich von der Verwendung in einem UFS-Profil, oder diese Schlüsselwörter gelten nur für ZFS-Profile.

- Einen Überblick über Schlüsselwörter für UFS- und ZFS-Profile finden Sie unter „Profilschlüsselwörter im Überblick“ auf Seite 116.
- In der folgenden Liste finden Sie alle Schlüsselwörter, die in einem ZFS-Profil verwendet werden können. Die Syntax ist für UFS- und ZFS-Profile gleich. Eine Beschreibung dieser Schlüsselwörter finden Sie in „Profilschlüsselwörter - Beschreibung und Beispiele“ auf Seite 118.
  - boot_device
  - cluster
  - dontuse
  - fdisk
  - filesys (Einhängen von Remote-Dateisystemen)
  - geo
  - locale
  - package
  - usedisk

bootenv-Profilschlüsselwort (ZFS und UFS)

Das Schlüsslwort `bootenv` legt die Eigenschaften der Bootumgebung fest. Eine Bootumgebung wird standardmäßig während der Installation mithilfe des Schlüsselworts `pool` erstellt. Wenn Sie das Schlüsselwort `bootenv` mit der Option `installbe` verwenden, können Sie die neue Bootumgebung benennen und innerhalb dieser Bootumgebung ein `/var`-Dataset erstellen.

Dieses Schlüsselwort kann in einem Profil zur Installation eines UFS-Dateisystems oder eines ZFS-Root-Pools verwendet werden.

- In einem UFS-Dateisystem dient dieses Schlüsselwort zum Erstellen einer leeren Bootumgebung zur späteren Installation eines Solaris Flash-Archivs. Eine vollständige Beschreibung des Schlüsselworts bootenv für UFS finden Sie unter „bootenv-Profilschlüsselwort (UFS und ZFS)“ auf Seite 128.
- Bei einem ZFS-Root-Pool ändert das Schlüsselwort `bootenv` die Eigenschaften der·während der Installation erstellten Bootumgebung. Diese Bootumgebung ist eine Kopie des Root-Dateisystems, das Sie installieren.

  Das Schlüsselwort `bootenv` kann mit den Optionen `installbe`, `bename` und `dataset` verwendet werden. Diese Optionen benennen die Bootumgebung und erstellen ein getrenntes `/var`-Dataset.

  ```
  bootenv installbe bename Neuer_BU_Name [dataset Einhängepunkt]
  ```

  `installbe` — Ändert die Eigenschaften der während der Installation erstellten Standard-Bootumgebung.

  `bename` — Legt den Namen der neu anzulegenden Boot-Umgebung fest, z.&;B. *Neuer_BU-Name* Dieser Name darf maximal 30 Zeichen lang sein und nur alphanumerische Zeichen enthalten. Multibyte-Zeichen sind nicht zulässig. Der Name muss auf dem System einmalig sein.

  `dataset` *Einhängepunkt* — Mit dem optionalen Schlüsselwort `dataset` können Sie ein vom ROOT-Dataset getrenntes `/var`-Dataset festlegen. Der Wert für *Einhängepunkt* ist auf `/var` beschränkt. Die `bootenv`-Syntaxzeile für ein getrenntes `/var` -Dataset würde beispielsweise wie folgt aussehen:

  ```
  bootenv installbe bename zfsroot dataset /var
  ```

Weitere Informationen zum Upgrade und Aktivieren einer Bootumgebung finden Sie in Kapitel 11, „Solaris Live Upgrade und ZFS (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

`install_type`-Schlüsselwort (ZFS und UFS)

Das Schlüsselwort `install_type` muss in jedem Profil enthalten sein. Für eine UFS-Installation stehen verschiedene Optionen·zur Verfügung. Das einzige für eine ZFS-Installation verfügbare Schlüsselwort ist `initial_install` . Diese Option installiert auf einem System ein neues Solaris-Betriebssystem. Die Profilsyntax ist wie folgt:

```
install_type initial_install
```

Hinweis – Die folgenden UFS-Optionen sind für ZFS-Installationen nicht verfügbar.

- `upgrade` - Sie müssen Solaris Live Upgrade zum Upgrade eines ZFS-Root-Pools verwenden. Siehe Kapitel 11, „Solaris Live Upgrade und ZFS (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.
- `flash_install` - Es können keine Solaris Flash-Archive installiert werden.
- `flash_update` - Es können keine Solaris Flash-Archive installiert werden.

`pool`-Profilschlüsselwort (nur ZFS)

Das Schlüsselwort pool definiert das neu zu erstellende Root-Pool. Das Pool wird dann mit einer durch das Schlüsselwort `cluster` angegebenen Softwaregruppe installiert. Zum Erstellen eines neuen Root-Pools werden die Optionen *Poolgröße*, *Swapgröße*, *Speicherabzugsgröße* und *Virtuelle_Geräteliste* benötigt.

`pool` *Poolname Poolgröße swapsize Speicherabzugsgröße Virtuelle_Geräteliste*

Poolname — Legt den Namen des neu anzulegenden Pools fest. Ein neues Pool wird mit der angegebenen *Poolgröße* sowie den angegebenen Geräten (*Virtuelle_Geräteliste*) erstellt.

Poolgröße — Kapazität des neu anzulegenden Pools. Wenn hier nur eine Zahl angegeben wird, nimmt das Schlüsselwört eine Kapazität in MB an, sofern nicht `g` (GB) angegeben wurde. Sie können auch die Option `auto` verwenden.

`auto` — Weist die im Rahmen der vorgegebenen Einschränkungen (wie z.&;B. Größe der Festplatten und beibehaltene·Festplattenbereiche) größtmögliche Poolkapazität zu

Hinweis – Die Bedeutung von `auto` für das Schlüsselwort *Poolgröße* unterscheidet sich von der Bedeutung des Schlüsselworts `filesys` für `auto` in UFS-Dateisystemen. In ZFS wird die Festplattenkapazität auf die erforderliche Mindestgröße hin überprüft. Wenn die Mindestgröße verfügbar ist, wird die im Rahmen der vorgegebenen Einschränkungen (wie z.&;B. Größe der Festplatten und beibehaltene·Festplattenbereiche) größtmögliche Poolkapazität zugewiesen.

Swapgröße Kapazität des Swap-Datenbereichs (`zvol`), der innerhalb eines neuen Root-Pools erstellt wird. Optionen: `auto` oder *Größe*.

- `auto` Die Kapazität des Swap-Bereichs wird automatisch festgelegt. Die Standardkapazität beträgt die Hälfte der Größe des physischen Speichers, jedoch mindestens 512&;MB und maximal 2&;GB. Mithilfe der Option `size` können Sie eine Kapazität außerhalb dieses Bereichs festlegen.
- *Größe* Dient zur Angabe einer Kapazität. Die Größe wird in MB angenommen, sofern nicht `g` (GB) angegeben ist.

Speicherabzugsgröße Kapazität des Speicherabzugsbereichs, der innerhalb eines neuen Root-Pools erstellt wird.

- `auto` Verwendung der Standardkapazität für den Swap-Bereich.
- *Größe* Dient zur Angabe einer Kapazität. Die Größe wird in MB angenommen, sofern nicht `g` (GB) angegeben ist.

Virtuelle_Geräteliste Ein oder mehrere Geräte, die zum Erstellen des Pools verwendet werden.

Geräte in *Virtuelle_Geräteliste* müssen Bereiche des Root-Pools sein. *Virtuelle_Geräteliste* kann entweder ein *einzelner Gerätename* im Format c*w*t*x*d *y*s*z* sein, oder es können die Optionen `mirror` bzw. `any` angegeben werden.

Hinweis – Das Format des Parameters *Virtuelle_Geräteliste* entspricht dem Format des Befehls `zpool create`.

Einzelner_Gerätename Ein Plattenbereich im Format c*w*t *x*d*y*s*z* wie z.&;B. `c0t0d0s0`.

`mirror` [*Gerätenamen*| `any`]	Legt die Datenspiegelungskonfiguration der Platte fest. Zurzeit werden Datenspiegelungskonfigurationen nur bei Verwendung mehrerer Datenträger unterstützt. Sie können beliebig viele Platten spiegeln; die Kapazität des angelegten Pools wird jedoch von der Kapazität der kleinsten angegebenen Platte festgelegt. Weitere Informationen zum Erstellen gespiegelter Speicherpools finden Sie unter „Speicher-Pools mit Datenspiegelung“ in *Solaris ZFS - Administrationshandbuch*. ■ *Gerätenamen* legt die Datenträger fest, die gespiegelt werden sollen. Diese Namen sind im Format c*w*t *x*d*y*s*z* (z.&;B. `c0t0d0s0` und `c0t0d1s5`) anzugeben. ■ Bei Angabe der Option `any` kann sich das Installationsprogramm verfügbare Datenträger aussuchen.
`any`	Das Installationsprogramm wählt einen geeigneten·Datenträger aus.

`root_device`-Profilschlüsselwort (ZFS und UFS)

`root_device` c*w*t*x*d*y*s*z*

`root_device` legt den Datenträger fest, der für das Root-Pool verwendet werden soll. Das Schlüsselwort `root_device` bestimmt, wo das Betriebssystem installiert wird. Der Gebrauch dieses Schlüsselworts ist (mit einigen Einschränkungen) für UFS- und ZFS-Dateisysteme gleich. Bei ZFS-Root-Pools ist das Root-Gerät auf ein einzelnes System beschränkt. Dieses Schlüsselwort eignet sich nicht für gespiegelte Pools.

c*w*t*x*d *y*s*z* Legt die Root-Platte fest, auf der das Betriebssystem installiert werden soll.

Weitere Ressourcen

Weitere Informationen zu den in diesem Kapitel enthaltenen Themen finden Sie in den in Tabelle 9–2 aufgeführten Ressourcen.

TABELLE 9–2 Weitere Ressourcen

Ressourcen	Speicherort
Informationen zu ZFS einschließl. Übersicht, Planung und schrittweise Anleitungen	*Solaris ZFS - Administrationshandbuch*
Liste aller JumpStart-Schlüsselwörter	Kapitel 8, „Benutzerdefinierte JumpStart-Installation (Referenz)“
Informationen zur Verwendung von Solaris Live Upgrade zum Migrieren von UFS auf ZFS sowie zum Erstellen einer neuen Boot-Umgebung in einem ZFS-Root-Pool	Kapitel 11, „Solaris Live Upgrade und ZFS (Überblick)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

TEIL II

Anhänge

Dieser Teil enthält Hinweise zur Fehlerbehebung sowie Referenzinformationen.

◆ ◆ ◆ ANHANG A

Fehlerbehebung (Vorgehen)

Dieses Kapitel enthält eine Liste spezifischer Fehlermeldungen und allgemeiner Probleme, die beim Installieren der Solaris 10 10/08-Software auftreten könnten. In diesem Kapitel wird außerdem erläutert, wie Sie Probleme beheben können. Die Erläuterungen in diesem Kapitel sind in die folgenden Abschnitte eingeteilt, je nachdem, wo im Installationsprozess das Problem auftrat.

- „Probleme beim Einrichten von Netzwerkinstallationen“ auf Seite 185
- „Probleme beim Booten eines Systems“ auf Seite 186
- „Neuinstallation von Solaris“ auf Seite 192
- „Upgrade von Solaris“ auf Seite 194

Hinweis – Der Text “bootable media” bezieht sich auf das Solaris-Installationsprogramm und die JumpStart-Installationsmethode.

Probleme beim Einrichten von Netzwerkinstallationen

`Unbekannter Client „`*Host-Name*`"`

Grund: Das Argument *Host-Name* im Befehl `add_install_client` bezieht sich nicht auf einen Host in diesem Naming Service.

Lösung: Fügen Sie den Host *Host-Name* zum Naming Service hinzu und führen Sie den Befehl `add_install_client` erneut aus.

`Fehler: <system name> ist in der NIS ethers Map nicht vorhanden`

`Fügen Sie ihn hinzu, und führen Sie den Befehl add_install_client erneut aus.`

Beschreibung: Beim Ausführen von `add_install_client` schlägt der Befehl mit der oben aufgeführten Fehlermeldung fehl.

Grund: Der Client, den Sie zum Installationsserver hinzufügen, ist in der Datei `/etc/ethers` des Servers nicht vorhanden.

Lösung: Fügen Sie die erforderlichen Informationen zur Datei `/etc/ethers` auf dem Installationsserver hinzu, und führen Sie den Befehl `add_install_client` erneut aus.

1. Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.
2. Suchen Sie die ethers-Adresse auf dem Client.

   ```
   # ifconfig -a grep ethers
   ether 8:0:20:b3:39:1d
   ```

3. Öffnen Sie die `/etc/ethers`-Datei in einem Editor auf dem Installationsserver. Fügen Sie die Adresse der Liste hinzu.
4. Führen Sie erneut den Befehl `add_install_client` auf dem Client aus.

   ```
   # ./add_install_client bluegill sun4u
   ```

Probleme beim Booten eines Systems

Booten von Medien, Fehlermeldungen

`le0: No carrier - transceiver cable problem`
Grund: Das System ist nicht mit dem Netzwerk verbunden.

Lösung: Handelt es sich hierbei um ein nicht vernetztes System, ignorieren Sie diese Meldung. Handelt es sich um ein vernetztes System, vergewissern Sie sich, dass die Ethernet-Verkabelung stimmt.

`Die gerade geladene Datei scheint nicht ausführbar zu sein`
Grund: Das System findet das richtige Medium zum Booten nicht.

Lösung: Überprüfen Sie, ob das System ordnungsgemäß für die Installation der Solaris 10 10/08-Software über das Netzwerk von einem Installationsserver aus eingerichtet wurde. Sie können beispielsweise folgende Überprüfungen durchführen:

- Wenn Sie die Abbilder der Solaris Operating System-DVD oder der Solaris Software-CDs auf den Installationsserver kopiert haben, vergewissern Sie sich, dass Sie bei der Einrichtung die richtige Plattformgruppe des Systems angegeben haben.
- Wenn Sie mit DVD oder CD vorgehen, vergewissern Sie sich, dass die Solaris Operating System-DVD bzw. Solaris Software-1-CD eingehängt und auf dem Installationsserver zugänglich ist.

`boot: cannot open <Dateiname> (nur SPARC-Systeme)`
Grund: Dieser Fehler tritt auf, wenn Sie den Speicherort von `boot -file` für das Booten überschreiben, indem Sie diesen explizit angeben.

Hinweis – *Dateiname* ist eine Variable für den Namen der betreffenden Datei.

Lösung: Gehen Sie wie folgt vor:

- Setzen Sie `boot -file` im PROM auf ? " (leer).
- Stellen Sie sicher, dass „diag-switch" auf „off" und auf „true" gesetzt ist.

`Kann von Datei/Gerät nicht booten`

Grund: Das Installationsmedium findet das Boot-Medium nicht.

Lösung: Stellen Sie sicher, dass folgende Bedingungen erfüllt sind:

- Das DVD-ROM- oder CD-ROM-Laufwerk muss ordnungsgemäß installiert und eingeschaltet sein.
- Die Solaris Operating System-DVD oder die Solaris Software-1-CD muss ins Laufwerk eingelegt sein.
- Der Datenträger ist unbeschädigt und nicht verschmutzt.

`WARNING: clock gained` *xxx* `days -- CHECK AND RESET DATE! (`**`nur SPARC-basierte Systeme`**`)`

Beschreibung: Diese Meldung dient zu Ihrer Information.

Lösung: Ignorieren Sie die Meldung und fahren Sie mit der Installation fort.

`Kein UFS-Dateisystem (`**`nur x86-basierte Systeme`**`)`

Grund: Bei der Installation von Solaris 10 10/08 (mit dem Solaris-Installationsprogramm oder benutzerdefiniertem JumpStart) wurde keine Boot-Diskette ausgewählt. Sie müssen jetzt das BIOS bearbeiten, um das System zu booten.

Lösung: Wählen Sie das BIOS, das gebootet werden soll. Erläuterungen dazu finden Sie in der BIOS-Dokumentation.

Booten von Medien, allgemeine Probleme

`Das System wird nicht gebootet.`

Beschreibung: Wenn Sie zum ersten Mal einen benutzerdefinierten JumpStart-Server einrichten, kann es beim Booten zu Problemen kommen, bei denen keine Fehlermeldung ausgegeben wird. Verwenden Sie den Boot-Befehl mit der Option `-v`, um Informationen über das System und den Boot-Vorgang abzurufen. Wenn Sie die Option `-v` verwenden, gibt der Boot-Befehl Fehlerbehebungsinformationen am Bildschirm aus.

Hinweis – Wenn diese Option nicht angegeben wird, werden die Meldungen zwar ausgegeben, aber die Ausgabe wird in die Systemprotokolldatei umgeleitet. Weitere Informationen entnehmen Sie bitte der Manpage `syslogd(1M)`.

Lösung: Für SPARC-Systeme geben Sie an der Eingabeaufforderung `ok` folgenden Befehl ein.

```
ok boot net -v - install
```

`Das Booten von DVD schlägt bei Systemen mit einem DVD-ROM-Laufwerk SD-M 1401 von Toshiba fehl.`

Beschreibung: Wenn das System mit einem DVD-ROM-Laufwerk SD-M1401 von Toshiba mit Firmware-Revision 1007 ausgestattet ist, kann das System nicht von der Solaris Operating System-DVD booten.

Lösung: Wenden Sie Patch 111649–03 oder später an, um die Firmware des Toshiba SD-M1401 DVD-ROM-Laufwerks zu aktualisieren. Das Patch 111649–03 ist auf sunsolve.sun.com verfügbar.

`Das System hängt sich auf oder eine Panik tritt auf, wenn Nicht-Speicher-PC-Karten eingelegt werden. (`**`Nur x86-basierte Systeme`**`)`

Grund: Nicht-Speicher-PC-Karten können nicht die gleichen Speicherressourcen verwenden wie andere Geräte.

Lösung: Um das Problem zu beheben, schlagen Sie in den Anweisungen zu Ihrer PC-Karte nach und überprüfen Sie den Adressbereich.

`Das System hängt sich auf, bevor die Systemeingabeaufforderung angezeigt wird. (`**`Nur x86-basierte Systeme`**`)`

Lösung: Es ist Hardware vorhanden, die nicht unterstützt wird. Lesen Sie hierzu in der Dokumentation des Hardware-Herstellers nach.

Booten vom Netzwerk, Fehlermeldungen

`WARNING: getfile: RPC failed: error 5 (RPC Timed out).`

Beschreibung: Dieser Fehler tritt auf, wenn zwei oder mehr Server in einem Netzwerk auf die Boot-Anforderung eines Installations-Clients reagieren. Der Installations-Client stellt eine Verbindung zum falschen Boot-Server her und die Installation hängt. Zu diesem Fehler kann es aus folgenden Gründen kommen:

Grund: *Grund;1:* Möglicherweise sind auf verschiedenen Servern `/etc/bootparams`-Dateien mit einem Eintrag für diesen Installationsclient vorhanden.

Lösung: *Grund 1:* Stellen Sie sicher, dass die Server im Netzwerk nicht mehrere `/etc/bootparams`-Einträge für den Installations-Client haben. Haben sie jedoch mehrere Einträge, entfernen Sie alle doppelten Client-Einträge in der Datei `/etc/bootparams` auf allen Installations- und Boot-Servern außer dem, den der Installationsclient verwenden soll.

Grund: *Grund 2:* Für den Installations-Client liegen möglicherweise mehrere `/tftpboot`- oder `/rplboot`-Verzeichniseinträge vor.

Lösung: *Grund 2:* Stellen Sie sicher, dass für den Installations-Client nicht mehrere `/tftpboot`- oder `/rplboot`-Verzeichniseinträge auf den Servern im Netzwerk vorliegen. Ist dies jedoch der Fall, entfernen Sie doppelte Client-Einträge aus den Verzeichnissen `/tftpboot` oder `/rplboot` auf allen Installations- und Boot-Servern außer auf dem, den der Installationsclient verwenden soll.

Grund: *Grund 3:* Möglicherweise liegt ein Installations-Client-Eintrag in der Datei `/etc/bootparams` auf einem Server und ein Eintrag in einer anderen Datei `/etc/bootparams` vor, der es allen Systemen ermöglicht, auf den Profilserver zuzugreifen. Ein solcher Eintrag sieht ungefähr folgendermaßen aus:

```
* install_config=profile_server:path
```

Dieser Fehler kann auch durch eine Zeile wie die oben genannte in der `bootparams`-Tabelle von NIS oder NIS+ verursacht werden.

Lösung: *Grund 3:* If a wildcard entry is in the naming service `bootparams` map or table (for example, `* install_config=`), delete it and add it to the `/etc/bootparams` file on the boot server.

`No network boot server. Unable to install the system. See installation instructions. (`**`nur SPARC-basierte Systeme`**`)`

Grund: Dieser Fehler tritt auf einem System auf, das Sie über das Netzwerk zu installieren versuchen. Das System ist nicht korrekt konfiguriert.

Lösung: Sorgen Sie dafür, dass das System korrekt für eine Installation über das Netzwerk eingerichtet wird. Siehe „Hinzufügen der über das Netzwerk zu installierenden Systeme mit einem CD-Abbild“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*.

`prom_panic: Could not mount file system (`**`nur SPARC-basierte Systeme`**`)`

Grund: Dieser Fehler tritt auf, wenn Sie Solaris über ein Netzwerk installieren, aber die Boot-Software Folgendes nicht finden kann:

- Solaris Operating System-DVD, entweder die DVD oder eine Kopie des DVD-Abbildes auf dem Installationsserver
- Abbild der Solaris Software-1-CD, entweder die Solaris Software-1-CD oder eine Kopie des CD-Abbildes auf dem Installationsserver.

Lösung: Vergewissern Sie sich, dass die Installationssoftware eingehängt und freigegeben ist.

- Bei der Installation von Solaris vom DVD-ROM- oder CD-ROM-Laufwerk des Installationsservers aus müssen Sie sicherstellen, dass die Solaris Operating System-DVD oder die Solaris Software-1-CD ins CD-ROM-Laufwerk eingelegt, eingehängt und in der Datei `/etc/dfs/dfstab` freigegeben ist.
- Bei der Installation von einer Kopie des Solaris Operating System-DVD-Abbildes oder des Solaris Software-1-CD-Abbildes auf der Festplatte des Installationsservers aus stellen Sie sicher, dass der Verzeichnispfad zu der Kopie in der Datei `/etc/dfs/dfstab` freigegeben ist.

`Timeout waiting for ARP/RARP packet...(`**`nur SPARC-basierte Systeme`**`)`

Grund: *Grund 1:* Der Client versucht, vom Netzwerk zu booten, kann aber kein System finden, das den Client kennt.

Lösung: *Grund 1:* Überprüfen Sie den Host-Namen des Systems im Naming Service NIS oder NIS+. Überprüfen Sie auch die `bootparams`-Suchreihenfolge in der Datei `/etc/nsswitch.conf` des Boot-Servers.

Beispielsweise bedeutet die folgende Zeile in der Datei `/etc/nsswitch.conf`, dass JumpStart oder das Solaris-Installationsprogramm zuerst in den NIS-Maps nach `bootparams`-Informationen sucht. Wenn das Programm keine Informationen findet, erfolgt eine Suche in der Datei `/etc/bootparams` auf dem Boot-Server.

```
bootparams: nis files
```

Grund: *Grund 2:* Die Ethernet-Adresse des Clients ist nicht korrekt.

Lösung: *Grund 2:* Stellen Sie sicher, dass die Ethernet-Adresse des Clients in der Datei `/etc/ethers` des Installationsservers korrekt angegeben ist.

Grund: *Grund 3:* In einer benutzerdefinierten JumpStart-Installation gibt der Befehl `add_install_client` die Plattformgruppe an, die einen bestimmten Server als Installationsserver verwendet. Wenn der falsche Architekturwert mit dem Befehl `add_install_client` verwendet wird, tritt dieses Problem auf. Beispiel: Der Rechner, der installiert werden soll, ist ein sun4u-Rechner, aber Sie haben i86pc verwendet.

Lösung: *Grund 3:* Führen Sie den Befehl `add_install_client` nochmals mit dem korrekten Architekturwert aus.

`IP: joining multicasts failed on tr0 - will use link layer broadcasts for multicast (`**`nur x86-basierte Systeme`**`)`

Grund: Diese Fehlermeldung erscheint beim Booten eines Systems mit einer Token Ring-Karte. Ethernet-Multicast und Token Ring-Multicast funktionieren nicht auf die gleiche Weise. Der Treiber gibt diese Fehlermeldung zurück, weil ihm eine ungültige Multicast-Adresse zur Verfügung gestellt wurde.

Lösung: Ignorieren Sie diese Fehlermeldung. Wenn Multicast nicht funktioniert, verwendet IP stattdessen Layer-Broadcasts, und die Installation schlägt deswegen nicht fehl.

`Requesting Internet address for` *Ethernet_Address* (**nur x86-basierte Systeme**)

Grund: Der Client versucht, vom Netzwerk zu booten, kann aber kein System finden, das den Client kennt.

Lösung: Überprüfen Sie, ob der Systemname im Naming Service enthalten ist. Wenn der Host-Name des Systems im Naming-Service NIS oder NIS+ aufgelistet ist und das System weiterhin diese Fehlermeldung ausgibt, versuchen Sie es mit einem Neustart.

`RPC: Timed out No bootparams (whoami) server responding; still trying...` (**nur x86-basierte Systeme**)

Grund: Der Client versucht, über das Netzwerk zu booten, aber er findet kein System mit einem Eintrag in der Datei `/etc/bootparams` auf dem Installationsserver.

Lösung: Geben Sie den Befehl `add_install_client` auf dem Installationsserver ein. Dieser Befehl fügt den entsprechenden Eintrag in die Datei `/etc/bootparams` ein und ermöglicht dem Client damit das Booten vom Netzwerk.

`Still trying to find a RPL server...` (**nur x86-basierte Systeme**)

Grund: Das System versucht, vom Netzwerk zu booten, aber der Server ist nicht so konfiguriert, dass er dieses System booten kann.

Lösung: Führen Sie auf dem Installationsserver für das zu installierende System `add_install_client` aus. Der Befehl `add_install_client` richtet ein Verzeichnis `/rplboot` ein, das das nötige Netzwerk-Boot-Programm enthält.

`CLIENT MAC ADDR: FF FF FF FF FF FF` (**nur Netzwerkinstallationen per DHCP**)

Grund: Der DHCP-Server ist nicht richtig konfiguriert. Dieser Fehler kann auftreten, wenn die Optionen oder Makros in der DHCP-Manager-Software nicht richtig definiert sind.

Lösung: Überprüfen Sie die Definition der Optionen und Makros in der DHCP-Manager-Software. Vergewissern Sie sich, dass die Router-Option definiert ist und den Wert für das Teilnetz aufweist, das bei der Netzwerkinstallation verwendet wird.

Booten vom Netzwerk, allgemeine Probleme

`Das System bootet über das Netzwerk, aber von einem anderen als dem angegebenen Installationsserver.`

Grund: Auf einem anderen System ist ein Eintrag in `/etc/bootparams` und eventuell auch in `/etc/ethers` für den Client enthalten.

Lösung: Aktualisieren Sie den Eintrag `/etc/bootparams` für das zu installierende System auf dem Namensserver. Der Eintrag muss folgende Syntax haben:

install_system `root=`*boot_server*`:`*path* `install=`*install_server*`:`*path*

Stellen Sie außerdem sicher, dass für den Installationsclient nur ein `bootparams`-Eintrag im Teilnetz vorliegt.

`Das System bootet nicht über das Netzwerk (`**`gilt nur für Installationen über das Netzwerk mit DHCP`**`).`

Grund: Der DHCP-Server ist nicht richtig konfiguriert. Dieser Fehler kann auftreten, wenn das System auf dem DHCP-Server nicht als Installationsclient konfiguriert wurde.

Lösung: Überprüfen Sie im DHCP Manager, dass für das betreffende Client-System Installationsoptionen und Makros definiert sind. Weitere Informationen finden Sie in „Vorkonfiguration der Systemkonfigurationsinformationen mit dem DHCP-Service (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Netzwerkbasierte Installation*.

Neuinstallation von Solaris

`Die Neuinstallation schlägt fehl.`

Lösung: Wenn die Solaris-Installation fehlschlägt, müssen Sie sie neu starten. Um die Installation neu zu starten, booten Sie das System von der Solaris Operating System-DVD, der Solaris Software-1-CD oder über das Netzwerk.

Sie können die Solaris-Software nicht deinstallieren, wenn sie teilweise installiert wurde. Sie müssen das System von einer Sicherungskopie wiederherstellen oder den Solaris-Installationsprozess erneut ausführen.

`/cdrom/cdrom0///SUNW `*xxxx*`/reloc.cpio: Gebrochene Pipe`

Beschreibung: Diese Fehlermeldung ist informativer Natur und hat keine Auswirkung auf die Installation. Die Bedingung tritt ein, wenn für einen Schreibzugriff auf ein Pipe kein Leseprozess vorhanden ist.

Lösung: Ignorieren Sie die Meldung und fahren Sie mit der Installation fort.

`WARNING: STANDARD-BOOT-GERÄT WECHSELN `**`(nur x86-basierte Systeme)`**

Grund: Diese Meldung dient zu Ihrer Information. Als Standard-Boot-Gerät ist im BIOS des Systems möglicherweise ein Gerät eingestellt, das zum Booten des Systems die Solaris Device Configuration Assistant erfordert.

Lösung: Fahren Sie mit der Installation fort und ändern Sie gegebenenfalls das Standard-Boot-Gerät des Systems, das im BIOS angegeben ist, nachdem Sie die Solaris-Software auf einem Gerät installiert haben, für das die Solaris Device Configuration Assistant nicht erforderlich ist.

x86 nur – Wenn Sie zum Testen eines benutzerdefinierten JumpStart-Profils für eine Erstinstallation das Schlüsselwort `locale` verwenden, schlägt der Test des Profils mithilfe des Befehls `pfinstall -D` fehl. Die Beschreibung einer Abhilfe finden Sie in der Erläuterung zur Fehlermeldung "could not select locale," im Abschnitt „Upgrade von Solaris“ auf Seite 194.

▼ x86: So überprüfen Sie eine IDE-Festplatte auf fehlerhafte Blöcke

IDE-Festplatten weisen fehlerhaften Blöcke nicht automatisch aus, wie andere von Solaris-Software unterstützte Festplatten. Bevor Sie Solaris auf einer IDE-Festplatte installieren, sollten Sie unter Umständen eine Oberflächenanalyse der Festplatte durchführen. Gehen Sie dazu folgendermaßen vor.

1. **Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.**

   Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

2. **Booten Sie vom Installationsdatenträger.**

3. **Wählen Sie Option 6 (Single User Shell), wenn Sie zur Auswahl einer Installationsart aufgefordert werden.**

4. **Siehe hierzu die Manpage `format`(1M).**

   ```
   # format
   ```

5. **Geben Sie das IDE-Laufwerk an, für das die Oberflächenanalyse durchgeführt werden soll.**

   ```
   # cxdy
   ```

   cx ist die Controller-Nummer

   d*y* ist die Gerätenummer

6. **Ermitteln Sie, ob eine `fdisk`-Partition vorhanden ist.**

   - Wenn bereits eine Solaris-Partition `fdisk` vorhanden ist, fahren Sie mit Schritt 7 fort.
   - Wenn noch keine Solaris-`fdisk`-Partition existiert, legen Sie mit dem Befehl `fdisk` eine Solaris-Partition auf der Festplatte an.

     ```
     format> fdisk
     ```

7 **Um die Oberflächenanalyse zu starten, geben Sie folgenden Befehl ein:**

```
format> analyze
```

8 **Um die aktuellen Einstellungen zu ermitteln, geben Sie folgenden Befehl ein:**

```
analyze> config
```

9 **(Optional) Wenn Sie die Einstellungen ändern wollen, geben Sie Folgendes ein:**

```
analyze> setup
```

10 **Um nach fehlerhaften Blöcken zu suchen, geben Sie Folgendes ein:**

```
analyze> Art_der_Oberflächenanalyse
```

Typ_der_Oberflächenanalyse kann „read“, „write“ oder „compare“ sein

Wenn `format` fehlerhafte Blöcke findet, weist es diese neu zu.

11 **Um die Analyse zu beenden, geben Sie Folgendes ein:**

```
analyze> quit
```

12 **Wollen Sie Blöcke zum erneuten Zuweisen angeben?**

- Wenn nicht, fahren Sie mit Schritt 13 fort.
- Wenn ja, geben Sie Folgendes ein:

```
format> repair
```

13 **Um das Programm format zu beenden, geben Sie folgendes ein:**

```
quit
```

14 **Booten Sie vom Datenträger im Mehrfachbenutzermodus durch Eingeben des folgenden Befehls.**

```
# exit
```

Upgrade von Solaris

Durchführen eines Upgrade, Fehlermeldungen

`No upgradable disks`
Grund: Ein Swap-Eintrag in der Datei `/etc/vfstab` verursacht das Fehlschlagen der Aktualisierung.

Lösung: Setzen Sie die folgenden Zeilen in der Datei `/etc/vfstab` auf Kommentar:

- Alle Swap-Dateien und -Speicherbereiche auf Platten, die nicht aufgerüstet werden
- Swap-Dateien, die nicht mehr vorhanden sind
- Nicht verwendete Swap-Bereiche

`usr/bin/bzcat not found`

Grund: Solaris Live Upgrade schlägt fehl, da ein benötigtes Patch-Cluster fehlt.

Lösung: Für die Installation von Solaris Live Upgrade ist ein Patch erforderlich. Eine stets aktuelle Patchliste finden Sie auf http://sunsolve.sun.com. Suchen Sie auf der SunSolve-Website nach dem Informationsdokument 72099.

`Es wurden aktualisierbare Solaris-Root-Geräte, jedoch keine geeigneten Partitionen für das Solaris-Installationsprogramm gefunden. Ein Upgrade mit dem Solaris-Installationsprogramm ist nicht möglich. Unter Umständen kann ein Upgrade mit der Solaris Software 1-CD durchgeführt werden. (Nur x86-basierte Systeme)`

Grund: Ein Upgrade mit der Solaris Software-1 ist nicht möglich, da nicht genug Platz vorhanden ist.

Lösung: Um ein Upgrade durchzuführen, können Sie entweder einen Swap-Bereich erstellen, das größer oder gleich 512 MB ist, oder ein anderes Upgrade-Verfahren verwenden, zum Beispiel das Solaris-Installationsprogramm von der Solaris Operating System-DVD, ein Netzwerk-Installationsabbild oder JumpStart.

`ERROR: Could not select locale (`**`nur x86-basierte Systeme`**`)`

Grund: Wenn Sie ein JumpStart-Profil mithilfe des Befehls `pfinstall -D` testen, schlägt der Dry Run-Test in den folgenden Situationen fehl:

- Das Profil enthält das Schlüsselwort “locale”.
- Sie testen ein Release, das GRUB-Software enthält **Ab Solaris-Release 10 1/06** erleichtert der GRUB-Bootloader das Booten unterschiedlicher Betriebssysteme mithilfe des GRUB-Menüs.

Mit der Einführung der GRUB-Software wurde die Miniroot komprimiert. Die Software findet deswegen in der komprimierten Miniroot nicht mehr die Liste der Gebietsschemata. Die Miniroot ist das kleinstmögliche Solaris root-Dateisystem (`/`). Sie befindet sich auf dem Solaris-Installationsdatenträger.

Lösung: Führen Sie die folgenden Schritte aus. Geben Sie die folgenden Werte ein:

- `MEDIA_DIR` is `/cdrom/cdrom0/`
- `MINIROOT_DIR` ist `$MEDIA_DIR /Solaris_10/Tools/Boot`
- `MINIROOT_ARCHIVE` is `$MEDIA_DIR /boot/x86.miniroot`

- TEMP_FILE_NAME is /tmp/test

1. Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.

   Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

2. Entpacken Sie das Miniroot-Archiv.

   ```
   # /usr/bin/gzcat $MINIROOT_ARCHIVE > $TEMP_FILE_NAME
   ```

3. Erstellen Sie mithilfe des Befehls lofiadm das Miniroot-Gerät.

   ```
   # LOFI_DEVICE=/usr/sbin/lofiadm -a $TEMP_FILE_NAME
   # echo $LOFI_DEVICE
   /dev/lofi/1
   ```

4. Hängen Sie die Miniroot mithilfe des Befehls lofi unter dem Miniroot-Verzeichnis ein.

   ```
   # /usr/sbin/mount -F ufs  $LOFI_DEVICE  $MINIROOT_DIR
   ```

5. Testen Sie das Profil.

   ```
   # /usr/sbin/install.d/pfinstall -D -c $MEDIA_DIR $path-to-jumpstart_profile
   ```

6. Hängen Sie nach dem Abschluss des Tests das lofi-Gerät wieder aus.he lofi device.

   ```
   # umount  $LOFI_DEVICE
   ```

7. Löschen Sie das lofi-Gerät.

   ```
   # lofiadm -d $TEMP_FILE_NAME
   ```

Durchführen eines Upgrade, allgemeine Probleme

Die Upgrade-Option wird nicht angeboten, obwohl auf dem System eine Upgrade-Version der Solaris-Software vorhanden ist.

Grund: *Grund 1:* Das Verzeichnis /var/sadm ist ein symbolischer Link oder wurde von einem anderen Dateisystem aus eingehängt.

Lösung: *Grund 1:* Verschieben Sie das Verzeichnis /var/sadm in das Root-Dateisystem (/) oder in das Dateisystem /var.

Grund: *Grund 2:* Die Datei /var/sadm/softinfo/INST_RELEASE fehlt.

Lösung: *Grund 2:* Erstellen Sie eine neue Datei INST_RELEASE. Verwenden Sie dazu folgende Vorlage:

```
OS=Solaris
VERSION=x
REV=0
```

x ist die Version der Solaris-Software auf dem System

Grund: *Grund 3:* SUNWusr ist in `/var/sadm/softinfo` nicht vorhanden.

Lösung: *Lösung 3:* Sie müssen eine Neuinstallation durchführen. Ein Upgrade der Solaris-Software ist nicht möglich.

`Der md-Treiber lässt sich nicht herunterfahren oder initialisieren.`

Lösung: Gehen Sie wie folgt vor:

- Handelt es sich bei dem Dateisystem nicht um ein RAID-1-Volume, so setzen Sie den entsprechenden Teil der Datei `vsftab` auf Kommentar.
- Handelt es sich um ein RAID-1-Volume, brechen Sie den Mirror-Verbund auf und führen Sie die Installation erneut durch. Informationen zum Aufbrechen des Mirror-Verbunds finden Sie in „Removing RAID-1 Volumes (Unmirroring)“ in *Solaris Volume Manager Administration Guide*.

`Das Upgrade schlägt fehl, weil das Solaris-Installationsprogramm ein Dateisystem nicht einhängen kann.`

Grund: Bei einem Upgrade wird durch das Skript versucht, alle in der Datei `/etc/vfstab` des Systems aufgeführten Dateisysteme in das Root-Dateisystem (`/`) einzuhängen, an dem das Upgrade durchgeführt wird. Wenn das Installationsskript ein Dateisystem nicht einhängen kann, schlägt es fehl und wird abgebrochen.

Lösung: Stellen Sie sicher, dass alle Dateisysteme in der Datei `/etc/vfstab` des Systems eingehängt werden können. Setzen Sie alle Dateisysteme in der Datei `/etc/vfstab` auf Kommentar, die nicht eingehängt werden können oder die das Problem anderweitig verursachen könnten, so dass das Solaris-Installationsprogramm beim Upgrade nicht versucht, sie einzuhängen. Systembasierte Dateisysteme jedoch, die zu aktualisierende Software enthalten (beispielsweise `/usr`), können nicht auf Kommentar gesetzt werden.

`Das Upgrade schlägt fehl.`

Beschreibung: Das System verfügt nicht über genügend Speicherplatz für das Upgrade.

Grund: Suchen Sie in „Upgrade mit Neuzuweisung von Festplattenspeicher“ in *Solaris 10 10/08 Installationshandbuch: Planung von Installationen und Upgrades* nach Informationen zum Speicherplatzproblem und versuchen Sie, ob Sie es ohne ein Auto-Layout zum Neuzuweisen des Speicherplatzes beseitigen können.

`Probleme beim Aktualisieren von RAID-1-Volumes als Root-Dateisysteme (/)`
Lösung: Sollten sich beim Upgrade mit Solaris Volume Manager RAID-1-Volumes als Root-Dateisystem (`/`) Probleme ergeben, schlagen Sie in Kapitel 25, „Troubleshooting Solaris Volume Manager (Tasks)“ in *Solaris Volume Manager Administration Guide* nach.

▼ So setzen Sie ein Upgrade nach einem Fehlschlag fort

Das Upgrade ist fehlgeschlagen und das System lässt sich nicht über die Software booten. Der Grund für den Fehlschlag liegt außerhalb Ihres Einflussbereichs, zum Beispiel ein Stromausfall oder der Ausfall einer Netzwerkverbindung.

1 **Starten Sie das System von der Solaris Operating System-DVD, der Solaris Software-1 CD oder über das Netzwerk neu.**

2 **Wählen Sie die Upgrade-Option für die Installation.**
Das Solaris-Installationsprogramm ermittelt, ob das System teilweise aufgerüstet wurde, und setzt das Upgrade fort.

x86: Probleme mit Solaris Live Upgrade bei der Verwendung von GRUB

Bei Verwendung von Solaris Live Upgrade und dem GRUB-Bootloader auf x86-basierten Systemen können die folgenden Fehler auftreten.

`ERROR: Das Tools-Installationsverzeichnis` *Installationsverzeichnis* `ist auf dem Produktmedium nicht vorhanden.`

`ERROR: The media` *Verzeichnis* `does not contain an operating system upgrade image.`
Beschreibung: Dieser Fehlermeldungen werden angezeigt, wenn mithilfe des Befehls `luupgrade` ein Upgrade auf eine neue Boot-Umgebung durchgeführt wird.

Grund: Es wird eine ältere Version Solaris Live Upgrade verwendet. Die auf Ihrem System installierten Packages von Solaris Live Upgrade sind mit dem Datenträger und dessen Softwareversion nicht kompatibel.

Lösung: Sie müssen stets die Solaris Live Upgrade-Packages des Releases verwenden, auf den Sie upgraden möchten.

Beispiel: Im folgenden Beispiel zeigt die Fehlermeldung an, dass die auf dem System installierten Packages von Solaris Live Upgrade nicht der auf dem Datenträger befindlichen Version entsprechen.

```
# luupgrade -u -n s10u1 -s /mnt
    Validating the contents of the media </mnt>.
    The media is a standard Solaris media.
    ERROR: The media product tools installation directory
</mnt/Solaris_10/Tools/Boot/usr/sbin/install.d/install_config> does
not exist.
    ERROR: The media </mnt> does not contain an operating system upgrade
image.
```

`ERROR: Cannot find or is not executable: </sbin/biosdev>.`

`ERROR: One or more patches required by Solaris Live Upgrade has not been installed.`

Grund: Eines oder mehrere, für Solaris Live Upgrade erforderliche Patches sind nicht auf Ihrem System installiert. Bitte beachten Sie, dass mit dieser Fehlermeldung nicht alle fehlenden Patches erkannt werden.

Lösung: Vor dem Arbeiten mit Solaris Live Upgrade müssen Sie erst alle erforderlichen Patches installieren. Eine stets aktuelle Patchliste finden Sie auf http://sunsolve.sun.com. Suchen Sie auf der SunSolve-Website nach dem Informationsdokument 72099.

`ERROR: Device mapping command </sbin/biosdev> failed. Please reboot and try again.`

Grund: *Grund 1:* Solaris Live Upgrade kann aufgrund vorheriger administrativer Aufgaben keine Geräte zuweisen.

Lösung: *Grund 1:* Booten Sie das System neu und starten Sie Solaris Live Upgrade erneut.

Grund: *Grund 2:* Wenn nach einem erneuten Booten des Systems diese Fehlermeldung wieder angezeigt wird, heißt das, dass sich im System zwei identische Festplatten befinden. Der Befehl zum Zuweisen von Geräten kann diese beiden Platten nicht unterscheiden.

Lösung: *Grund 2:* Erstellen Sie auf einer der beiden Festplatten eine `fdisk`-Dummypartition. Weitere Informationen entnehmen Sie bitte der Manpage `fdisk`(1M) Booten Sie dann das System neu.

`Cannot delete the boot environment that contains the GRUB menu`

Grund: Solaris Live Upgrade besitzt die Einschränkung, dass Boot-Umgebungen, die das GRUB-Menü enthalten, nicht gelöscht werden können.

Lösung: Mit den Befehlen `lumake`(1M) oder `luupgrade`(1M) können Sie diese Boot-Umgebung wiederverwenden.

`Das Dateisystem, das das GRUB-Menü enthält, wurde versehentlich neu angelegt, die Festplatte besitzt jedoch immer noch die gleichen Speicherbereiche wie zuvor. So kann es beispielsweise sein, dass die Festplatte nicht neu partitioniert wurde.`

Grund: Das Dateisystem, das das GRUB-Menü enthält, wird zum Booten des Systems benötigt. Solaris Live Upgrade-Befehle zerstören das GRUB-Menü nicht. Wenn das GRUB-Menü jedoch mit anderen, nicht zu Solaris Live Upgrade gehörenden Befehlen versehentlich oder aus anderen Gründen überschrieben bzw. zerstört wird, versucht die Software zur Wiederherstellung des Systems, das GRUB-Menü neu zu installieren. Diese Software kopiert das GRUB-Menü beim nächsten Booten des Systems in das gleiche Dateisystem. Es kann zum Beispiel sein, dass Sie mit den Befehlen `newfs` oder `mkfs` das GRUB-Menü versehentlich zerstört haben. Damit das GRUB.Menü ordnungsgemäß wiederhergestellt werden kann, muss der Speicherbereich die folgenden Bedingungen erfüllen:

- Es muss ein einhängbares Dateisystem besitzen.
- Es muss zur gleichen Boot-Umgebung von Solaris Live Upgrade wie vorher gehören.

Nehmen Sie vor einem Systemneustart am Speicherbereich die erforderlichen Korrekturen vor.

Lösung: Starten Sie das System neu. Es wird automatisch eine Sicherungskopie des GRUB-Menüs installiert.

`The GRUB menu's menu.lst file was accidentally deleted.`

Lösung: Starten Sie das System neu. Es wird automatisch eine Sicherungskopie des GRUB-Menüs installiert.

▼ Systempanik bei einem Upgrade mit Solaris Live Upgrade und Veritas VxVm

Wenn Sie bei einem Upgrade Solaris Live Upgrade benutzen und Veritas VxVM läuft, kommt es beim Neustart zu einer Systempanik. Um diese zu vermeiden, müssen Sie das Upgrade mit dem folgenden Verfahren durchführen. Das Problem tritt auf, wenn Packages nicht den neuen Solaris-Richtlinien für Packages entsprechen.

1 Melden Sie sich als Superuser an oder nehmen Sie eine entsprechende Rolle an.

Rollen umfassen Autorisierungen und privilegierte Befehle. Weitere Informationen zu Rollen finden Sie unter „Configuring RBAC (Task Map)“ in *System Administration Guide: Security Services*.

2 Erstellen Sie eine inaktive Boot-Umgebung. Siehe „Erstellen einer neuen Boot-Umgebung“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

3 Vor dem Upgrade der inaktiven Boot-Umgebung müssen Sie in der inaktiven Boot-Umgebung die vorhandene Veritas-Software deaktivieren.

a. Hängen Sie die inaktive Boot-Umgebung ein.

```
# lumount Name_der_inaktiven_Bootumgebung Einhängepunkt
```

Beispiel:

```
# lumount solaris8 /mnt
```

b. Wechseln Sie in das Verzeichnis, das die `vfstab` enthält. Beispiel:

```
# cd /mnt/etc
```

c. Erstellen Sie eine Kopie der Datei `vfstab` der inaktiven Boot-Umgebung. Beispiel:

```
# cp vfstab vfstab.501
```

d. Setzen Sie in der kopierten Datei `vfstab` alle Veritas-Dateisystemeinträge auf Kommentar. Beispiel:

```
#  sed  '/vx\/dsk/s/^/#/g'  < vfstab >  vfstab.novxfs
```

Als erstes Zeichen erscheint in den entsprechenden Zeilen ein #. Dadurch gelten diese Zeilen als Kommentarzeilen. Beachten Sie, dass diese Kommentarzeilen sich von den Kommentarzeilen in der Systemdatei unterscheiden.

e. Kopieren Sie die geänderte Datei `vfstab`. Beispiel:

```
# cp vfstab.novxfs vfstab
```

f. Wechseln Sie in das Verzeichnis mit der Systemdatei der inaktiven Boot-Umgebung. Beispiel:

```
# cd /mnt/etc
```

g. Erstellen Sie eine Kopie der Systemdatei der inaktiven Boot-Umgebung. Beispiel:

```
# cp system system.501
```

h. Kennzeichnen Sie alle Einträge des Typs "forceload:" als Kommentare, die `drv/vx` enthalten.

```
# sed '/forceload:   drv\/vx/s/^/*/' <system> system.novxfs
```

Als erstes Zeichen erscheint in den entsprechenden Zeilen ein *. Dadurch gelten diese Zeilen als Befehlszeilen. Beachten Sie, dass diese Kommentarzeilen sich von den Kommentarzeilen in der Datei `vfstab` unterscheiden.

i. Erstellen Sie die Veritas-Datei `install-db`. Beispiel:

```
# touch vx/reconfig.d/state.d/install-db
```

j. Hängen Sie die inaktive Boot-Umgebung aus.

```
# luumount Name_der_inaktiven_Bootumgebung
```

4 Führen Sie das Upgrade der inaktiven Boot-Umgebung durch. Siehe Kapitel 5, „Ausführen eines Upgrades mit Solaris Live Upgrade (Vorgehen)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

5 Aktivieren Sie die inaktive Boot-Umgebung. Siehe „Aktivieren einer Boot-Umgebung“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*.

6 Fahren Sie das System herunter.

```
# init 0
```

7 Booten Sie die inaktive Boot-Umgebung im Einzelbenutzermodus:

```
OK boot -s
```

Mehrere Meldungen und Fehlermeldungen, die „vxvm“ oder „VXVM“ enthalten, werden angezeigt. Ignorieren Sie diese. Die inaktive Boot-Umgebung wird aktiv.

8 Führen Sie ein Upgrade von Veritas durch.

a. Entfernen Sie das Package Veritas VRTSvmsa vom System. Beispiel:

```
# pkgrm VRTSvmsa
```

b. Wechseln Sie in das Verzeichnis mit den Veritas-Packages.

```
# cd /Speicherort_der_Veritas_Software
```

c. Fügen Sie die neuesten Veritas-Packages zum System hinzu:

```
#  pkgadd -d ‘pwd‘ VRTSvxvm VRTSvmsa VRTSvmdoc VRTSvmman VRTSvmdev
```

9 Stellen Sie die ursprüngliche Datei `vfstab` und die ursprünglichen Systemdateien wieder her:

```
# cp /etc/vfstab.original /etc/vfstab
# cp /etc/system.original /etc/system
```

10 Starten Sie das System neu.

```
# init 6
```

x86: Service-Partition wird auf Systemen ohne bereits vorhandene Service-Partition nicht standardmäßig erzeugt

Wenn Sie Solaris 10 10/08 auf einem System installieren, das noch keine Service- bzw. Diagnosepartition enthält, wird eine solche unter Umständen nicht automatisch vom Installationsprogramm erzeugt. Wenn auf der Festplatte mit der Solaris-Partition auch eine Service-Partition enthalten sein soll, dann müssen Sie die Service-Partition vor der Installation von Solaris 10 10/08 neu erstellen.

Bei der Installation von Solaris 8 2/02 auf einem System mit Service-Partition behält das Installationsprogramm die Service-Partition u. U. nicht bei. Sofern Sie das Layout der Boot-Partition `fdisk` nicht manuell bearbeiten, um die Service-Partition beizubehalten, wird die Service-Partition vom Installationsprogramm gelöscht.

Hinweis – Wenn Sie die Service-Partition bei der Installation von Solaris 8 2/02 nicht ausdrücklich beibehalten haben, dann ist es u. U. nicht möglich, die Service-Partition wiederherzustellen und ein Upgrade auf Solaris 10 10/08 durchzuführen.

Um auf der Festplatte mit der Solaris-Partition auch eine Service-Partition einzurichten, wählen Sie eine der nachfolgenden Problemlösungen.

▼ So installieren Sie die Software von einem Netzwerk-Installationsabbild oder der Solaris Operating System-DVD

Zur Installation von einem Netzwerk-Installationsabbild oder von der Solaris Operating System-DVD über das Netzwerk gehen Sie wie folgt vor:

1. **Löschen Sie den Inhalt der Festplatte.**

2. **Legen Sie vor der Installation die Service-Partition an. Verwenden Sie hierzu die Diagnose-CD für Ihr System.**

   Wie Sie die Service-Partition erzeugen, entnehmen Sie bitte der Dokumentation zur jeweiligen Hardware.

3. **Booten Sie das System über das Netzwerk.**

   Der Bildschirm für die Anpassung der `fdisk`-Partitionen wird angezeigt.

4 **Um das Standard-Layout für die Bootplatten-Partitionen zu laden, klicken Sie auf „Default".**

Das Installationsprogramm behält die Service-Partition bei und erzeugt die Solaris-Partition.

▼ So installieren Sie von der Solaris Software-1-CD oder einem Netzwerk-Installationsabbild

Zur Installation von der Solaris Software-1-CD oder von einem Netzwerkinstallationsabbild auf einem Boot-Server mithilfe des Solaris-Installationsprogramms gehen Sie wie folgt vor:

1 **Löschen Sie den Inhalt der Festplatte.**

2 **Legen Sie vor der Installation die Service-Partition an. Verwenden Sie hierzu die Diagnose-CD für Ihr System.**

Wie Sie die Service-Partition erzeugen, entnehmen Sie bitte der Dokumentation zur jeweiligen Hardware.

3 **Das Installationsprogramm fordert Sie dazu auf, eine Methode zur Erstellung der Solaris-Partition auszuwählen.**

4 **Booten Sie das System.**

5 **Wählen Sie die Option** `Use rest of disk for Solaris partition`**.**

Das Installationsprogramm behält die Service-Partition bei und erzeugt die Solaris-Partition.

6 **Schließen Sie die Installation ab.**

◆ ◆ ◆ ANHANG B

Zusätzliche SVR4-Packaging-Anforderungen (Referenz)

Dieser Anhang richtet sich an Systemadministratoren, die mit dem benutzerdefinierten JumpStart-Programm oder Solaris Live Upgrade Packages installieren bzw. entfernen müssen, insbesondere Packages von Drittherstellern. Die Beachtung dieser Packaging-Anforderungen bewirkt:

- Das aktuell ausgeführte System wird nicht verändert. Sie können also ein Upgrade mit Solaris Live Upgrade durchführen und nicht-globale Zonen sowie Diskless-Clients erzeugen und warten.
- Es wird verhindert, dass Packages bei der Verwendung von Installationsprogrammen wie der benutzerdefinierten Jump-Start-Installation interaktiv ausgeführt werden; d. h. die automatisierte Installation ohne Benutzereingriff ist möglich.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Verhindern einer Modifikation des aktuellen BS“ auf Seite 205
- „Verhindern der Benutzerinteraktion bei Installation oder Upgrade“ auf Seite 209
- „Einstellen von Package-Parametern für Zonen“ auf Seite 211

Verhindern einer Modifikation des aktuellen BS

Wenn Sie die in diesem Abschnitt beschriebenen Anforderungen erfüllen, bleibt das aktuell ausgeführte BS unverändert.

Verwenden absoluter Pfade

Für eine erfolgreiche Installation eines Betriebssystems müssen die Packages alternative Root-Dateisysteme (/) wie z. B. eine inaktive Solaris Live Upgrade-Boot-Umgebung erkennen und fehlerfrei behandeln.

Packages können in der Datei pkgmap (Package-Map) absolute Pfade enthalten. Sind die Dateien vorhanden, werden sie relativ zu dem Verzeichnis geschrieben, das mit der Option -R des Befehls pkgadd angegeben wird. Packages, die absolute und relative (verschiebbare) Pfade enthalten, können ebenfalls in einem alternativen Root-Dateisystem (/) installiert werden. $PKG_INSTALL_ROOT wird absoluten und verschiebbaren Dateien vorangestellt, so dass alle Pfade bei der Installation mit pkgadd korrekt aufgelöst werden.

Verwenden des Befehls pkgadd -R

Packages, die mit dem Befehl pkgadd und der Option -R installiert bzw. mit dem Befehl pkgrm und der Option -R entfernt werden, dürfen das zurzeit laufende System nicht modifizieren. Dieses Leistungsmerkmal kommt in der benutzerdefinierten JumpStart-Installation, in Solaris Live Upgrade, nicht-globalen Zonen und Diskless-Clients zum Einsatz.

Prozedurskripten, die in den mit dem Befehl pkgadd und der Option -R installierten bzw. mit dem Befehl pkgrm und der Option -R entfernten Packages enthalten sind, dürfen das zurzeit laufende System nicht modifizieren. Von Ihnen zur Verfügung gestellte Installationsskripten müssen alle Verzeichnisse und Dateien mit vorangestellter $PKG_INSTALL_ROOT-Variable referenzieren. Das Package muss alle Verzeichnisse und Dateien mit dem vorangestellten $PKG_INSTALL_ROOT-Präfix schreiben. Das Package darf keine Verzeichnisse ohne $PKG_INSTALL_ROOT-Präfix entfernen.

Tabelle B–1 zeigt Beispiele der Skriptsyntax.

TABELLE B–1 Beispiele für Installationskriptsyntax

Skripttyp	Richtige Syntax	Falsche Syntax
Auszüge aus "if"-Anweisungen (Borne-Shell)	`if [ -f ${PKG_INSTALL_ROOT}\ /etc/myproduct.conf ] ; then`	`if [ -f /etc/myproduct.conf ] ; \ then`
Entfernen einer Datei	`/bin/rm -f ${PKG_INSTALL_ROOT}\ /etc/myproduct.conf`	`/bin/rm -f /etc/myproduct.conf`
Ändern einer Datei	`echo "test=no" > ${PKG_INSTALL_ROOT}\ /etc/myproduct.conf`	`echo "test=no" > \ /etc/myproduct.conf`

Unterschiede zwischen $PKG_INSTALL_ROOT und $BASEDIR - Übersicht

`$PKG_INSTALL_ROOT` ist der Speicherort des Root-Dateisystems (`/`) auf dem Rechner, zu dem Sie das Package hinzufügen. Der Speicherort wird auf das `-R`-Argument des Befehls `pkgadd` gesetzt. So wird beispielsweise beim Aufruf des folgenden Befehls `$PKG_INSTALL_ROOT` während der Package-Installation zu `/a`.

```
# pkgadd -R /a SUNWvxvm
```

`$BASEDIR` verweist auf das *verschiebbare* Basisverzeichnis, in dem verschiebbare Package-Objekte installiert werden. Hier werden nur verschiebbare Objekte installiert. Nicht verschiebbare Objekte (Objekte mit *absoluten* Pfaden in der Datei `pkgmap`) werden immer relativ zur inaktiven Boot-Umgebung installiert, nicht jedoch relativ zum aktuellen `$BASEDIR`. Wenn ein Package keine verschiebbaren Objekte aufweist, wird es als absolutes bzw. nicht verschiebbares Package bezeichnet. `$BASEDIR` ist nicht definiert und steht Package-Prozedurskripten nicht zur Verfügung.

Angenommen, die Datei `pkgmap` eines Packages enthält zwei Einträge:

```
1 f none sbin/ls 0555 root sys 3541 12322 1002918510
1 f none /sbin/ls2 0555 root sys 3541 12322 2342423332
```

In der Datei `pkginfo` ist `$BASEDIR` definiert:

```
BASEDIR=/opt
```

Bei Installation dieses Packages mit dem folgenden Befehl wird `ls` in `/a/opt/sbin/ls`, aber `ls2` als `/a/sbin/ls2` installiert.

```
# pkgadd -R /a SUNWtest
```

Richtlinien zum Schreiben von Skripten

Package-Prozedurskripten müssen vom aktuell ausgeführten BS unabhängig sein, damit eine Änderung des BS verhindert werden kann. Prozedurskripten definieren Aktionen, die an bestimmten Punkten während der Installation bzw. der Deinstallation von Packages auftreten. Mit diesen vordefinierten Namen können vier Prozedurskripten erstellt werden: `preinstall`, `postinstall`, `preremove` und `postremove`.

TABELLE B–2 Richtlinien zum Erstellen von Skripten

Richtlinien	Betrifft Solaris Live Upgrade	Betrifft nicht-globale Zonen
Skripten müssen in der Bourne-Shell (`/bin/sh`) geschrieben werden. Die Bourne-Shell wird beim Ausführen von Prozedurskripten vom Befehl `pkgadd` als Interpreter verwendet.	X	X
Skripten dürfen keine Prozesse starten oder stoppen und dürfen nicht von der Ausgabe von Befehlen wie `ps` oder `truss` abhängig sein, die vom Betriebssystem abhängen und Informationen über das zurzeit laufende System zurückgeben.	X	X
In Skripten können andere Standard-UNIX-Befehle wie `expr`, `cp` und `ls` sowie weitere Befehle verwendet werden, die das Schreiben von Skripten erleichtern.	X	X
Packages müssen auf allen unterstützten Solaris-Versionen lauffähig sein. Daher müssen alle Befehle, die von einem Skript aufgerufen werden, in allen unterstützten Versionen zur Verfügung stehen. Befehle, die in späteren Versionen als Solaris 8 eingeführt oder entfernt wurden, dürfen also nicht verwendet werden. Um herauszufinden, ob ein bestimmter Befehl bzw. eine Option in Solaris 8, 9 oder 10 unterstützt wird, können Sie im jeweiligen *Solaris Reference Manual AnswerBook* dieser Version auf http://docs.sun.com nachschlagen.	X	

Erhalten der Diskless-Client-Kompatibilität

Packages dürfen keine Befehle ausführen, die vom Package selbst geliefert werden. Dadurch wird die Diskless-Client-Kompatibilität gewährleistet und sichergestellt, dass keine Befehle ausgeführt werden, für die gemeinsam genutzte Bibliotheken benötigt werden, die noch nicht installiert sind.

Überprüfen von Packages

Alle Packages müssen mit `pkgchk` validiert werden. Nachdem Sie ein Package erstellt haben, müssen Sie es vor der Installation mit dem folgenden Befehl überprüfen:

 # pkgchk -d *Verz_name Pkg-Name*

Verz_name Gibt den Namen des Verzeichnisses an, in dem sich das Package befindet

Pkg-Name Gibt den Namen des Packages an

BEISPIEL B–1 Testen von Packages

Nachdem Sie ein Package erstellt haben, müssen Sie es testen, indem Sie es mit der Option `R -Verz_name` des Befehls *pkgadd* in einem alternativen Root-Dateisystem (`/`) installieren. Nach der Installation des Packages ist es wie in diesem Beispiel mit dem Befehl `pkgchk` auf Fehler zu überprüfen.

BEISPIEL B–1 Testen von Packages *(Fortsetzung)*

```
# pkgadd -d . -R /a SUNWvxvm
# pkgchk -R /a SUNWvxvm
```

Es sollten keine Fehler angezeigt werden.

BEISPIEL B–2 Testen eines Packages in `/export/SUNWvxvm`

Wenn ein Package in `/export/SUNWvxvm` gespeichert ist, führen Sie den folgenden Befehl aus:

```
# pkgchk -d /export SUNWvxvm
```

Es sollten keine Fehler angezeigt werden.

Beim Erstellen, Bearbeiten und Löschen von Dateien können andere Befehle das Package überprüfen. Die folgenden Befehle sind ein Beispiel hierfür.

- Mit dem Befehl `dircmp` und `fssnap` können Sie zum Beispiel verifizieren, ob sich Packages wie gewünscht verhalten.
- Mit dem Befehl `ps` können Sie außerdem die Konformität von Dämonen testen, indem Sie sicherstellen, dass das Package keine Dämonen stoppt oder startet.
- Mit den Befehlen `truss`, `pkgadd -v` und `pkgrm` können Sie testen, ob die Konformität der Package-Installation zur Laufzeit gegeben ist, doch dies funktioniert möglicherweise nicht in allen Situationen. Im folgenden Beispiel entfernt der Befehl `truss` alle schreibgeschützten Nicht-`$TEMPDIR`-Zugriffe und zeigt nur die nicht schreibgeschützten Zugriffe auf Pfade an, die nicht in der angegebenen inaktiven Boot-Umgebung liegen.

```
# TEMPDIR=/a; export TEMPDIR
# truss -t open /usr/sbin/pkgadd -R ${TEMPDIR} SUNWvxvm \
2>&1 > /dev/null | grep -v O_RDONLY | grep -v \
'open("'${TEMPDIR}
```

Verhindern der Benutzerinteraktion bei Installation oder Upgrade

Bei der Verwendung der folgenden Solaris-Standarddienstprogramme müssen Packages hinzugefügt oder entfernt werden, ohne dass der Benutzer zu Eingaben aufgefordert wird:

- Benutzerdefiniertes JumpStart-Programm
- Solaris Live Upgrade
- Programm Solaris-Installationsprogramm

- Solaris-Zonen

Um zu testen, dass sich ein Package ohne Benutzereingriffe installieren lässt, können Sie mit dem Befehl `pkgadd` und der Option `-a` eine neue Administrationsdatei einrichten. Die Option `-a` definiert eine Installationsadministrationsdatei, die anstelle der Standardadministrationsdatei verwendet wird. Bei Verwendung der Standarddatei wird der Benutzer möglicherweise zur Eingabe weiterer Informationen aufgefordert. Sie können eine Administrationsdatei erstellen, in der `pkgadd` angewiesen wird, diese Abfragen auszulassen und das Package ohne Bestätigung seitens des Benutzers zu installieren. Genaueres hierzu entnehmen Sie bitte der Manpage `admin(4)` oder `pkgadd(1M)`.

An den folgenden Beispielen wird deutlich, wie der Befehl `pkgadd` die Administrationsdatei verwendet.

- Wenn keine Administrationsdatei zur Verfügung gestellt wird, verwendet `pkgadd` die Datei `/var/sadm/install/admin/default`. Dabei werden jedoch möglicherweise Benutzereingriffe erforderlich.

  ```
  # pkgadd
  ```

- Wenn Sie über die Befehlszeile eine relative Administrationsdatei angeben, sucht `pkgadd` in `/var/sadm/install/admin` nach dem Dateinamen. In diesem Beispiel lautet der Name der relativen Administrationsdatei `nocheck`, und `pkgadd` sucht nach `/var/sadm/install/admin/nocheck`.

  ```
  # pkgadd -a nocheck
  ```

- Wenn eine absolute Datei angegeben wird, verwendet `pkgadd` diese. In diesem Beispiel sucht `pkgadd` in `/tmp` nach der Administrationsdatei `nocheck`.

  ```
  # pkgadd -a /tmp/nocheck
  ```

BEISPIEL B–3 Installations-Administrationsdatei

Sie sehen hier ein Beispiel für eine Installations-Administrationsdatei, die im Zusammenhang mit dem Dienstprogramm `pkgadd` nur sehr wenig Benutzerinteraktion erfordert. Sofern das Package nicht mehr Festplattenspeicher benötigt, als auf dem System verfügbar ist, greift `pkgadd` auf diese Datei zu und installiert das Package, ohne den Benutzer zur Eingabe von Informationen aufzufordern.

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
space=ask
setuid=nocheck
```

BEISPIEL B–3 Installations-Administrationsdatei *(Fortsetzung)*

```
confiict=nocheck
action=nocheck
basedir=default
```

Einstellen von Package-Parametern für Zonen

Pakete enthalten Parameter, mit denen festgelegt wird, wie ihr Inhalt verteilt wird und in einem System mit installierten nicht-globalen Zonen sichtbar gemacht wird. Die Paketparameter `SUNW_PKG_ALLZONES`, `SUNW_PKG_HOLLOW` und `SUNW_PKG_THISZONE` definieren die Eigenschaften von Paketen auf einem System mit installierten Zonen. Diese Parameter sind so einzustellen, dass Packages auf einem System mit nicht-globalen Zonen verwaltet werden können.

In der folgenden Tabelle sind vier gültige Kombinationen für das Einstellen von Paketparameter aufgeführt. In dieser Tabelle nicht aufgeführte Kombinationen sind unzulässig und haben zur Folge, das Packages nicht ordnungsgemäß installiert werden.

Hinweis – Achten Sie darauf, alle drei Paketparameter einzustellen. Sie können alle drei Paketparameter frei lassen. Die Package-Tools interpretieren leer gelassene Package-Parameter für Zonen als Wert “false”; es wird jedoch davon abgeraten, Package-Parameter nicht einzustellen. Durch das Einstellen aller drei Paketparameter geben Sie das exakte Verhalten der Paket-Tools beim Installieren oder Deinstallieren eines Pakets vor.

TABELLE B–3 Zulässige Einstellungen für Package-Parameter auf Systemen mit Zonen

SUNW_PKG_ALLZONES Einstellung	SUNW_PKG_HOLLOW Einstellung	SUNW_PKG_THISZONE Einstellung	Paketbeschreibung
false	false	false	Dies ist die Standardeinstellung für Pakete, die nicht für alle Zonen-Paketparameter Werte angeben. Ein Paket mit diesen Einstellungen kann entweder in der globalen oder in einer nicht-globalen Zone installiert werden. ■ Wenn der Befehl `pkgadd` in der globalen Zone ausgeführt wird, erfolgt die Installation des Pakets in der globalen Zone und in allen nicht-globalen Zonen. ■ Wenn der Befehl `pkgadd` in einer nicht-globalen Zone ausgeführt wird, erfolgt die Installation des Pakets nur in der nicht-globalen Zone. In beiden Fällen ist der gesamte Inhalt des Pakets in allen Zonen sichtbar, in denen das Paket installiert wurde.
false	false	true	Ein Paket mit diesen Einstellungen kann entweder in der globalen oder in einer nicht-globalen Zone installiert werden. Wenn nach der Installation neue nicht-globale Zonen erstellt werden, wird das Paket nicht auf diese neuen nicht-globalen Zonen installiert. ■ Wenn der Befehl `pkgadd` in der globalen Zone ausgeführt wird, erfolgt die Installation des Pakets nur in der globalen Zone. ■ Wenn der Befehl `pkgadd` in einer nicht-globalen Zone ausgeführt wird, erfolgt die Installation des Pakets nur in der nicht-globalen Zone. In beiden Fällen ist der gesamte Inhalt des Pakets in der Zone sichtbar, in der das Paket installiert wurde.
true	false	false	Ein Paket mit diesen Einstellungen kann nur in der globalen Zone installiert werden. Wenn der Befehl `pkgadd` ausgeführt wird, erfolgt die Installation des Pakets in der globalen Zone und in allen nicht-globalen Zonen. Der gesamte Inhalt des Pakets ist in allen Zonen sichtbar. **Hinweis –** Jeder Versuch, das Paket in einer nicht-globalen Zone zu installieren, schlägt fehl.

TABELLE B–3 Zulässige Einstellungen für Package-Parameter auf Systemen mit Zonen *(Fortsetzung)*

SUNW_PKG_ALLZONES Einstellung	SUNW_PKG_HOLLOW Einstellung	SUNW_PKG_THISZONE Einstellung	Paketbeschreibung
true	true	false	Ein Paket mit diesen Einstellungen kann nur vom globalen Administrator und nur in der globalen Zone installiert werden. Wenn der Befehl `pkgadd` ausgeführt wird, erfolgt die Installation des Paketinhalts vollständig in der globalen Zone. Wenn die Paketparameter eines Pakets auf diese Werte gesetzt sind, wird der Paketinhalt nicht an eine nicht-globale Zone geliefert. In den nicht-globalen Zonen werden lediglich die Paket-Installationsinformationen installiert, die für die Anzeige des Pakets erforderlich sind. Dies ermöglicht die Installation anderer Pakete, deren Installation von diesem Paket abhängt. Weitere Informationen zu diesen sog. „hohlen" Packages finden Sie in Kapitel 24, „Allgemeine Informationen zu den Paketen und Patches auf einem Solaris-System mit installierten Zonen" in *Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones* Für die Paketabhängigkeitsprüfung scheint das Paket in allen Zonen installiert zu sein. ■ In der globalen Zone ist der gesamte Inhalt des Pakets sichtbar. ■ In nicht-globalen Whole Root Zones ist der gesamte Inhalt des Pakets nicht sichtbar. ■ Wenn eine nicht-globale Zone ein Dateisystem von der globalen Zone übernimmt, ist ein in diesem Dateisystem installiertes Paket auch in einer nicht-globalen Zone sichtbar. Alle anderen Dateien, die aus dem Paket stammen, sind innerhalb der nicht-globalen Zone nicht sichtbar. Beispielsweise kann eine nicht-globale Sparse Root Zone bestimmte Verzeichnisse gemeinsam mit der globalen Zone nutzen. Diese Verzeichnisse sind schreibgeschützt. Nicht-globale Sparse Root Zones nutzen unter anderem das Dateisystem `/platform` gemeinsam mit der globalen Zone. Ein weiteres Beispiel sind Pakete mit Dateien, die nur für die Boot-Hardware relevant sind. **Hinweis –** Jeder Versuch, das Paket in einer nicht-globalen Zone zu installieren, schlägt fehl.

Beschreibung	Weitere Informationen
Ausführliche Informationen zu Packages und Zonen	Kapitel 24, „Allgemeine Informationen zu den Paketen und Patches auf einem Solaris-System mit installierten Zonen“ in *Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones*
Übersicht über kleine und ganze root-Zonen	Kapitel 16, „Einführung in Solaris Zones“ in *Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones*
Informationen zu Package-Eigenschaften und Parametern	`pkginfo(4)`
Informationen zum Anzeigen von Werten von Package-Parametern	`pkgparam(1)`

Hintergrundinformationen

An den folgenden Stellen finden Sie Hintergrundinformationen zu Packaging-Anforderungen und spezifischer Befehlssyntax.

Für spezifischere Informationen zu Packaging-Anforderungen sowie Begriffsdefinitionen siehe	Kapitel 6, „Advanced Techniques for Creating Packages“ in *Application Packaging Developer's Guide*
Für grundlegende Informationen zum Hinzufügen und Entfernen von Packages und zur Installations-Administrationsdatei siehe	Kapitel 19, „Managing Software (Overview)“ in *System Administration Guide: Basic Administration*
Ausführliche Informationen zu bestimmten, in diesem Anhang erwähnten Befehlen finden Sie in den Manpages	`dircmp(1)`, `fssnap(1M)`, `ps(1)`, oder `truss(1)` `pkgadd(1M)`, `pkgchk(1M)`, oder `pkgrm(1M)`
Einen Überblick zu Solaris Live Upgrade bietet	Kapitel 2, „Solaris Live Upgrade (Übersicht)“ in *Solaris 10 10/08 Installationshandbuch: Solaris Live Upgrade und Planung von Upgrades*
Einen Überblick zum benutzerdefinierten JumpStart-Verfahren finden Sie in	Kapitel 2, „Benutzerdefinierte JumpStart-Installation (Übersicht)“
Einen Überblick zu Zones bietet	Kapitel 16, „Einführung in Solaris Zones“ in *Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones*

Glossar

3DES ([Dreifach-DES] Triple-Data Encryption Standard, Standard für die dreifache Datenverschlüsselung). Eine symmetrische Verschlüsselungsmethode, die eine Schlüssellänge von 168 Bit bietet.

Abgeleitetes Profil Ein Profil, das bei einer benutzerdefinierten JumpStart-Installation dynamisch von einem Begin-Skript erstellt wird.

Abgesichertes Boot-Archiv **Nur x86**: Ein Boot-Archiv, das zur Wiederherstellung verwendet wird, falls das primäre Boot-Archiv beschädigt ist. Dieses Boot-Archiv startet das System, ohne das Root-Dateisystem (`/`) einzuhängen. Im GRUB-Menü wird dieses Boot-Archiv als „abgesichert" bezeichnet. Es dient hauptsächlich dazu, das primäre Boot-Archiv neu zu erzeugen (also das Boot-Archiv, mit dem das System normalerweise gestartet wird). Siehe *Boot-Archiv*.

AES (Advanced Encryption Standard) Eine symmetrische 128-Bit-Blockdaten-Verschlüsselungstechnik. Die U.S.-Regierung hat die Rijndael-Variante des Algorithmus im Oktober 2000 als Verschlüsselungsstandard angenommen. AES ersetzt die DES-Verschlüsselung als Regierungsstandard.

Aktualisierung Eine Installation, bei der bereits auf dem System vorhandene Software desselben Typs geändert wird. Im Gegensatz zu einem Upgrade (einer Aufstufung) kann eine Aktualisierung (engl. Update) auch eine Herabstufung des Systems bewirken. Anders als bei einer Erst- bzw. Neuinstallation, muss Software desselben Typs wie die zu installierende Software bereits auf dem System vorhanden sein, damit eine Aktualisierung vorgenommen werden kann.

Archiv Eine Datei, die einen Satz von Dateien enthält, die von einem Mastersystem kopiert wurden. Die Datei enthält auch Identifikationsinformationen über das Archiv, zum Beispiel einen Namen und das Datum der Archiverstellung. Nach der Installation eines Archivs auf einem System verfügt dieses System über genau dieselbe Konfiguration wie das Mastersystem.

Dabei kann es sich auch um ein Differenzarchiv handeln, d. h. ein Solaris Flash-Archiv, das nur die Unterschiede zwischen zwei Systemabbildern (einem unveränderten und einem aktualisierten Master-Abbild) enthält. Ein Differenzarchiv enthält die auf dem Klonsystem beizubehaltenden, zu ändernden oder zu löschenden Dateien. Eine solche differentielle Aktualisierung ändert nur die angegebenen Dateien und kann nur auf Systeme angewendet werden, deren Software mit derjenigen des unveränderten Master-Abbilds übereinstimmt.

Aufgabe Eine benutzerdefinierte Aufgabe, die ein Computersystem ausführen soll.

Aushängen Das Beenden des Zugriffs auf ein Verzeichnis auf einer Festplatte, die mit einem lokalen Rechner oder mit einem Remote-Rechner in einem Netzwerk verbunden ist.

Bedienfeld Ein 'Behälter', in dem der Inhalt eines Fensters, Dialogfeldes oder Applets angeordnet ist. In einem Bedienfeld werden möglicherweise Benutzereingaben aufgenommen und bestätigt. Häufig wird in Assistenten eine Folge mehrere Bedienfelder angezeigt, die den Benutzer durch einen bestimmten Vorgang leiten.

Befehlszeile Eine Zeichenkette, die mit einem Befehl beginnt, oft gefolgt von Argumenten einschließlich Optionen, Dateinamen und anderen Ausdrücken, und mit einem Zeilenendezeichen endet.

Begin-Skript Ein benutzerdefiniertes Bourne-Shell-Skript, spezifiziert innerhalb der Datei `rules`, das bestimmte Aufgaben ausführt, bevor die Solaris-Software auf dem System installiert wird. Begin-Skripte können ausschließlich bei benutzerdefinierten JumpStart-Installationen eingesetzt werden.

Benutzerdefinierte JumpStart-Installation Ein Installationstyp, bei dem die Solaris-Software auf der Basis eines benutzerdefinierten Profils automatisch auf einem System installiert wird. Man kann benutzerdefinierte Profile von Benutzern und Systemen erstellen. Eine benutzerdefinierte JumpStart-Installation ist eine JumpStart-Installation, die Sie erstellen.

Benutzerdefinierte probes-Datei Eine Datei, die sich im gleichen JumpStart-Verzeichnis befinden muss wie die `rules`-Datei. Es handelt sich dabei um ein Bourne-Shell-Skript, das zwei Typen von Funktionen enthält: Probe-Funktionen (Sondierfunktionen) und Comparison-Funktionen (Vergleichsfunktionen). Probe-Funktionen sammeln die Informationen, die benötigt werden, oder setzen eine entsprechende `SI_`-Umgebungsvariable, die Sie definieren. Probe-Funktionen werden zu Probe-Schlüsselwörtern. Comparison-Funktionen rufen die entsprechende Probe-Funktion auf, vergleichen die Ausgabe der Probe-Funktion und geben 0 zurück, wenn das Schlüsselwort übereinstimmt, bzw. 1, wenn das Schlüsselwort nicht übereinstimmt. Comparison-Funktionen werden zu Rule-Schlüsselwörtern. Siehe auch *rules-Datei*.

Betriebssystemserver Ein System, das den Systemen in einem Netzwerk Dienste zur Verfügung stellt. Um Diskless Clients bedienen zu können, benötigt ein Betriebssystemserver Festplattenkapazitäten, die für die Root-Dateisysteme (`/`) und Swap-Bereiche der einzelnen Diskless Clients vorgesehen sind (`/export/root`, `/export/swap`).

Boot-Archiv **Nur x86**: Ein Boot-Archiv ist ein Satz grundlegender Systemdateien, die zum Booten von Solaris dienen. Diese Dateien werden beim Systemstart benötigt, bevor das Root-Dateisystem (`/`) eingehängt wird. Auf jedem System werden zwei Boot-Archive vorgehalten:

- Das Boot-Archiv, das zum Booten von Solaris verwendet wird. Dieses Archiv wird auch als „primäres“ Boot-Archiv bezeichnet.
- Das Boot-Archiv, das zur Wiederherstellung verwendet wird, falls das primäre Boot-Archiv beschädigt ist. Dieses Boot-Archiv startet das System, ohne das Root-Dateisystem (`/`) einzuhängen. Im GRUB-Menü wird dieses Boot-Archiv als „abgesichert“ bezeichnet. Es dient hauptsächlich dazu, das primäre Boot-Archiv neu zu erzeugen (also das Boot-Archiv, mit dem das System normalerweise gestartet wird).

Boot-Loader
Nur x86: Der Boot-Loader ist das erste Programm, das nach dem Einschalten eines Systems ausgeführt wird. Dieses Programm leitet den Boot-Vorgang ein.

Boot-Server
Ein Serversystem, das den Clientsystemen in Teilnetzen des gleichen Netzwerks die Programme und Daten zur Verfügung stellt, die diese zum Starten benötigen. Ein Boot-Server ist bei einer Installation über das Netzwerk erforderlich, wenn sich der Installationsserver in einem anderen Teilnetz befindet als die Systeme, auf denen die Solaris-Software installiert werden soll.

Boot-Umgebung
Eine Sammlung obligatorischer Dateisysteme (Festplatten-Speicherbereiche und Einhängepunkte), die Voraussetzung für die Ausführung des Betriebssystems Solaris sind. Diese Festplatten-Speicherbereiche können sich auf einer Festplatte befinden oder über mehrere Festplatten verteilt sein.

Die aktive Boot-Umgebung ist die zum jeweiligen Zeitpunkt gebootete. Es kann immer nur eine aktive Boot-Umgebung gebootet sein. Eine inaktive Boot-Umgebung ist zum jeweiligen Zeitpunkt nicht gebootet, kann sich aber in einem Wartezustand befinden und auf Aktivierung beim nächsten Systemneustart warten.

Booten
Laden der Systemsoftware in den Hauptspeicher und starten dieser Software.

`bootlog-cgi`-**Programm**
Das CGI-Programm, das es einem Webserver ermöglicht, während einer WAN-Boot-Installation die Meldungen zum Booten von Remote-Clients sowie die Installationskonsolen-Meldungen aufzunehmen und zu speichern.

CD
Optischer Datenträger (im Gegensatz zu einem magnetischen Datenträger), der die auf dem CD-Markt übliche Schreibung erkennt. Bei CD-ROMs und DVD-ROMs handelt es sich z. B. um optische Datenträger.

certificate authority (Zertifizierungsstelle)
(ZA, auch Zertifizierungsstelle) Eine vertrauenswürdige Fremdorganisation oder -firma, die digitale Zertifikate zum Zweck der Erstellung von digitalen Signaturen und Paaren öffentlicher und privater Schlüssel ausstellt. Der ZA garantiert, dass der Benutzer, für den ein eindeutiges Zertifikat ausgestellt wurde, wirklich ist, wer er/sie zu sein behauptet.

`certstore`-**Datei**
Eine Datei, die ein digitales Zertifikat für ein spezifisches Clientsystem enthält. Während einer SSL-Aushandlung wird der Client möglicherweise aufgefordert, dem Server diese Zertifikatdatei vorzulegen. Anhand dieser Datei verifiziert der Server die Client-Identität.

CGI
(Common Gateway Interface) Eine Schnittstelle, über die externe Programme mit dem HTTP-Server kommunizieren. Programme, die auf die Verwendung der CGI ausgerichtet sind, werden als CGI-Programme oder CGI-Skripten bezeichnet. CGI-Programme verarbeiten Formulare oder parsen Ausgaben, die der Server normalerweise nicht verarbeitet oder parst.

checksum
Das Ergebnis der Addition einer Gruppe von Datenelementen. Der Vorgang dient zum Überprüfen der Gruppe. Bei den Datenelementen kann es sich um Zahlen oder um andere Zeichenfolgen handeln. Auch diese werden bei der Prüfsummenberechnung wie Zahlen behandelt. Anhand des Prüfsummenwertes kann sichergestellt werden, dass die Kommunikation zwischen zwei Geräten erfolgreich war.

Client Im Client-Server-Kommunikationsmodell ist der Client ein Prozess, der von fern auf Ressourcen auf einem Rechenserver zugreift, zum Beispiel auf Verarbeitungsleistung oder auf eine große Hauptspeicherkapazität.

Cluster Eine logische Sammlung von Packages (Softwaremodulen). Die Solaris-Software ist in mehrere *Softwaregruppen* eingeteilt, die jeweils aus Clustern und *Packages* bestehen.

Datei `rules.ok` Eine generierte Version der `rules`-Datei. Die Datei `rules.ok` wird von der benutzerdefinierten JumpStart-Installationssoftware dazu benötigt, ein System einem Profil zuzuordnen. Zum Erstellen der Datei `rules.ok` *muss* das Skript `check` verwendet werden.

Dateiserver Ein Server, der als Speicher für die Software und die Dateien für die Systeme in einem Netzwerk dient.

Datensatz Ein generischer Name für die folgenden ZFS-Entitäten: Klone, Dateisysteme, Momentaufnahmen oder Volumes.

decryption (Entschlüsselung) Der Vorgang, bei dem kodierte Daten in Normaltext konvertiert werden. Siehe auch encryption (Verschlüsselung).

DES (Data Encryption Standard) Eine 1981 entwickelte und 3 als ANSI X.92.56 von ANSI standardisierte symmetrische Verschlüsselungsmethode. DES verwendet einen 56-Bit-Schlüssel.

DHCP (Dynamic Host Configuration Protocol) Ein Protokoll der Anwendungsschicht (Application Layer). Ermöglicht es einzelnen Computern bzw. Clients in einem TCP/IP-Netzwerk, eine IP-Adresse oder andere Netzwerkkonfigurationsinformationen von einem oder mehreren designierten und zentral gepflegten DHCP-Servern zu extrahieren. Die verringert den Aufwand für die Pflege und Verwaltung großer IP-Netzwerke.

Dienstprogramm Ein Standardprogramm, das beim Kauf eines Computers in der Regel im Preis inbegriffen ist und für verschiedene interne Funktionen des Computers zuständig ist.

Differenzarchiv Ein Solaris Flash-Archiv, das nur die Unterschiede zwischen zwei Systemabbildern, einem unveränderten und einem aktualisierten Master-Abbild, enthält. Ein Differenzarchiv enthält die auf dem Klonsystem beizubehaltenden, zu ändernden oder zu löschenden Dateien. Eine solche differentielle Aktualisierung ändert nur die angegebenen Dateien und kann nur auf Systeme angewendet werden, deren Software mit derjenigen des unveränderten Master-Abbilds übereinstimmt.

Digitales Zertifikat Eine nicht übertragbare, unfälschbare digitale Datei, die von einer Stelle ausgestellt wurde, die für beide Kommunikationspartner bereits als vertrauenswürdig gilt.

Diskless Client Ein Client in einem Netzwerk, der als Festplattenspeicher einen Server benötigt.

Dokument-Root-Verzeichnis Der Ursprung einer Hierarchie auf einem Webserver, auf dem sich die Dateien, Grafiken und Daten befinden, die Sie den auf den Webserver zugreifenden Benutzern zur Verfügung stellen möchten.

Domäne Ein Teil der Namenshierarchie im Internet. Eine Domäne ist eine Gruppe von Systemen in einem lokalen Netzwerk, die Administrationsdateien gemeinsam nutzen.

Domänenname — Der Name einer Gruppe von Systemen in einem lokalen Netzwerk, die Administrationsdateien gemeinsam nutzen. Der Domänenname ist erforderlich, damit der Network Information Service (NIS) ordnungsgemäß funktioniert. Ein Domain-Name besteht aus einer Folge von Komponentennamen, die durch Punkte getrennt sind (Beispiel: `tundra.mpk.ca.us`). Der Domänenname wird von links nach rechts gelesen. Weiter links stehen also die Komponentennamen von übergeordneten (und in der Regel über Netzwerk zugänglichen) administrativen Bereichen.

Einhängen — Der Zugriff auf ein Verzeichnis von einer Festplatte aus, die mit einem Rechner verbunden ist, welcher die Einhängeanforderung absetzt, oder von einer Remote-Festplatte in einem Netzwerk aus. Zum Einhängen eines Dateisystems ist ein Einhängepunkt auf dem lokalen System erforderlich und der Name des einzuhängenden Dateisystems muss bekannt sein (zum Beispiel `/usr`).

Einhängepunkt — Ein Workstation-Verzeichnis, in das ein Dateisystem eingehängt wird, das auf einem Remote-Rechner residiert.

encryption (Verschlüsselung) — Der Vorgang, bei dem Daten unverständlich gemacht werden, um sie vor unberechtigten Zugriffen zu schützen. Die Verschlüsselung basiert auf einem Code, dem Schlüssel (key), mit dem die Daten wieder entschlüsselt werden. Siehe auch decryption (Entschlüsselung).

Erstinstallation / Neuinstallation — Eine Installation, bei der die aktuell installierte Software überschrieben oder eine leere Festplatte initialisiert wird.

Mit einer Neu- bzw. Erstinstallation des Solaris-BS wird die Festplatte (bzw. mehrere) des Systems mit der neuen Version des Solaris-BS überschrieben. Wenn das Solaris-BS nicht auf dem System läuft, müssen Sie eine Neuinstallation ausführen. Wenn eine upgrade-fähige Version des Solaris-BS auf dem System läuft, wird bei einer Neuinstallation die Festplatte überschrieben und weder das BS noch lokale Änderungen werden beibehalten.

`/etc/netboot`-Verzeichnis — Das Verzeichnis auf einem WAN-Boot-Server, in dem sich die für eine WAN-Boot-Installation erforderlichen Client-Konfigurationsinformationen und Sicherheitsdaten befinden.

`/etc`-Verzeichnis — Ein Verzeichnis mit wichtigen Systemkonfigurationsdateien und Wartungsbefehlen.

`/export`-Dateisystem — Ein Dateisystem auf einem Betriebssystemserver, das mit anderen Systemen im Netzwerk gemeinsam genutzt wird. Das Dateisystem `/export` zum Beispiel kann das Root-Dateisystem (`/`) und den Swap-Bereich für Diskless Clients sowie die Home-Verzeichnisse für Benutzer im Netzwerk enthalten. Diskless Clients benötigen das Dateisystem `/export` auf einem Betriebssystemserver, damit sie booten und laufen können.

Fallback — Das System greift auf die Umgebung zurück, die zuvor ausgeführt wurde. Ein Fallback ist erforderlich, wenn Sie eine Boot-Umgebung aktivieren und die Boot-Umgebung, mit der gebootet werden soll, fehlschlägt oder ein unerwünschtes Verhalten zeigt.

fdisk-Partition Eine logische Partition auf einem Festplattenlaufwerk bei x86-basierten Systemen, die für ein bestimmtes Betriebssystem vorgesehen ist. Zum Installieren der Solaris-Software muss auf einem x86-basierten System mindestens eine `fdisk`-Partition eingerichtet werden. Bei x86-basierten Systemen sind bis zu vier verschiedene `fdisk`-Partitionen pro Festplatte zulässig. Diese Partitionen können einzelne Betriebssysteme aufnehmen. Jedes Betriebssystem muss sich in einer eindeutigen `fdisk`-Partition befinden. Ein System kann nur eine Solaris `fdisk`-Partition pro Festplatte aufnehmen.

Festplatte Magnetischer Datenträger, bestehend aus einer runden Platte oder Gruppe von Platten, eingeteilt in konzentrische Spuren und Sektoren. Dient zum Speichern von Daten, zum Beispiel in Dateien. Siehe auch CD (optischer Datenträger).

Festplatten-konfigurationsdatei Eine Datei, die die Struktur einer Festplatte angibt (z. B. Byte/Sektor, Flags, Speicherbereiche). Festplattenkonfigurationsdateien ermöglichen die Verwendung des Befehls `pfinstall`, um von einem einzelnen System aus Profile auf Festplatten unterschiedlicher Größe zu testen.

Dateisystem Im Betriebssystem SunOS™ ein Netzwerk von Dateien und Verzeichnissen in einer Baumstruktur, auf die zugegriffen werden kann.

Finish-Skript Ein benutzerdefiniertes Bourne-Shell-Skript, angegeben in der `rules`-Datei, das Aufgaben ausführt, nachdem die Solaris-Software auf dem System installiert wurde, aber bevor das System neu gestartet wird. Finish-Skripten werden bei benutzerdefinierten JumpStart-Installationen eingesetzt.

format Daten in eine bestimmte Struktur bringen oder eine Festplatte in Sektoren aufteilen, so dass darauf Daten gespeichert werden können.

Funktionstasten Die mindestens 10 Tasten auf der Tastatur mit der Bezeichnung F1, F2, F3 usw., denen bestimmte Funktionen zugeordnet sind.

Gemeinsam genutzte Dateisysteme Dateisysteme, bei denen es sich um benutzerdefinierte Dateien handelt, zum Beispiel `/export/home` und `/swap`. Diese Dateisysteme werden von der aktiven und der inaktiven Boot-Umgebung gemeinsam genutzt, wenn Sie Solaris Live Upgrade verwenden. Gemeinsam genutzte Dateisysteme enthalten in der aktiven und der inaktiven Boot-Umgebung den gleichen Einhängepunkt in der Datei `vfstab`. Eine Aktualisierung der gemeinsam genutzten Dateien in der aktiven Boot-Umgebung bewirkt gleichzeitig auch eine Aktualisierung der Daten in der inaktiven Boot-Umgebung. Gemeinsame genutzte Dateisysteme werden standardmäßig gemeinsam genutzt. Sie können jedoch einen Zielspeicherbereich angeben. Daraufhin werden die Dateisysteme kopiert.

Gesamte Solaris-Softwaregruppe Eine Softwaregruppe, die die vollständige Solaris-Version enthält.

Gesamte Solaris-Softwaregruppe plus OEM-Unterstützung Eine Softwaregruppe, die das vollständige Solaris-Version plus zusätzliche Hardwareunterstützung für OEMs enthält. Diese Softwaregruppe ist zu empfehlen, wenn die Solaris-Software auf SPARC-Servern installiert werden soll.

Globale Zone In Solaris Zones gilt die globale Zone sowohl als Standardzone des Systems als auch als Zone für die systemweite Administrationssteuerung. Die globale Zone ist die einzige Zone, von der aus eine nicht-globale Zone konfiguriert, installiert, verwaltet oder deinstalliert werden kann. Die Verwaltung der Systeminfrastruktur, wie beispielsweise physische Geräte, das Routing oder die dynamische Rekonfiguration (DR), ist nur in der globalen Zone möglich. Entsprechend privilegierte Prozesse, die in der globalen Zone ausgeführt werden, können auf Objekte zugreifen, die anderen Zonen zugewiesen sind. Siehe auch *Solaris Zones* und *Nicht-globale Zone*.

GRUB **Nur x86**: Der GNU GRand Unified Bootloader (GRUB) ist ein Open-Source-Boot-Loader mit einer einfachen Menüoberfläche. Das Menü zeigt eine Liste mit den Betriebssystemen, die auf dem betreffenden System installiert sind. Über GRUB lassen sich diese unterschiedlichen Betriebssysteme (z. B. Solaris, Linux oder Microsoft Windows) komfortabel starten.

GRUB-Bearbeitungsmenü **Nur x86**: Ein Boot-Menü, das dem GRUB-Hauptmenü untergeordnet ist. Es enthält verschiedene GRUB-Befehle. Mit diesem Befehlen lässt sich das Boot-Verhalten anpassen.

GRUB-Hauptmenü **Nur x86**: Ein Boot-Menü mit der Liste der Betriebssysteme, die auf dem betreffenden System installiert sind. Über dieses Menü können Sie komfortabel ein bestimmtes Betriebssystem starten, ohne dafür die `fdisk`-Partitionseinstellungen oder die BIOS-Konfiguration ändern zu müssen.

Hard Link Ein Verzeichniseintrag, der auf eine Datei auf einer Festplatte verweist. Mehrere dieser Verzeichniseinträge können auf die gleiche physische Datei verweisen.

hash Eine Zahl, die aus einer Eingabe generiert wird und wesentlich kürzer ist als diese Eingabe. Für identische Eingaben wird stets derselbe Ausgabewert generiert. Hash-Funktionen lassen sich in Tabellensuchalgorithmen, bei der Fehlersuche und Manipulationserkennung einsetzen. Für die Manipulationserkennung werden die Hash-Funktionen so gewählt, dass es unwahrscheinlich ist, dasselbe Hash-Ergebnis für zwei Eingaben zu erhalten. MD5 und SHA-1 sind Beispiele für Einweg-Hash-Funktionen. Beispielsweise reduziert ein Meldungs-Digest eine Eingabe variabler Länge auf einen kleinen Wert.

Hashing Der Vorgang, bei dem eine aus Buchstaben bestehende Zeichenkette in einen Wert oder Schlüssel umgeformt wird, der die ursprüngliche Zeichenkette darstellt.

HMAC Verschlüsselte Hashing-Methode zur Nachrichtenauthentifizierung. HMAC wird mit einer iterativen kryptografischen Hash-Funktion wie MD5 oder SHA-1 zusammen mit einem geheimen gemeinsam genutzten Schlüssel verwendet. Die kryptografischen Stärke von HMAC hängt von den Eigenschaften der zu Grunde liegenden Hash-Funktion ab.

Host-Name Der Name, unter dem ein System den anderen Systemen im Netzwerk bekannt ist. Dieser Name muss unter den Systemen in einer Domain (in der Regel bedeutet das innerhalb einer Organisation) eindeutig sein. Ein Host-Name kann aus einer beliebigen Kombination von Buchstaben, Ziffern und Minuszeichen (-) bestehen, kann aber nicht mit einem Minuszeichen beginnen oder enden.

HTTP (Hypertext Transfer Protocol) (n.) Das Internet-Protokoll, das Hypertext-Objekte von Remote-Hosts abruft. Dieses Protokoll basiert auf TCP/IP.

HTTPS Eine sichere Version von HTTP, die unter Verwendung von SSL (Secure Sockets Layer) implementiert wird.

Installationsserver Ein Server, der die Solaris-DVD- oder -CD-Abbilder zur Verfügung stellt, von denen andere Systeme in einem Netzwerk Solaris installieren können (auch bekannt als *Medienserver*). Sie können einen Installationsserver erstellen, indem Sie die Solaris-DVD- bzw. -CD-Abbilder auf die Serverfestplatte kopieren.

IPv6 IPv6 ist eine Version (Version 6) des Internet Protocol (IP), die einen Entwicklungsschritt über die aktuelle Version IPv4 (Version 4) hinaus darstellt. Die Bereitstellung von IPv6 mithilfe definierter Umsetzungsmechanismen unterbricht den aktuellen Systembetrieb nicht. Darüber hinaus liefert IPv6 eine Plattform für eine neue Internet-Funktionalität.

JumpStart-Installation Ein Installationstyp, bei dem die Solaris-Software automatisch auf einem System installiert wird, und zwar mithilfe der werkseitig installierten JumpStart-Software.

JumpStart-Verzeichnis Bei benutzerdefinierten JumpStart-Installationen von einer Profildiskette entspricht das JumpStart-Verzeichnis dem Root-Verzeichnis auf der Diskette, das alle wichtigen, benutzerdefinierten JumpStart-Dateien enthält. Bei benutzerdefinierten JumpStart-Installationen von einem Profilserver entspricht das JumpStart-Verzeichnis dem Verzeichnis auf dem Server, das alle wichtigen, benutzerdefinierten JumpStart-Dateien enthält.

Kerberos Ein Netzwerkauthentisierungsprotokoll, das es mithilfe einer leistungsstarken Kryptographie mit geheimen Schlüsseln Clients und Servern ermöglicht, einander über eine nicht abgesicherte Netzwerkverbindung zuverlässig zu identifizieren.

`keystore`-**Datei** Eine Datei, in der sich die von Client und Server gemeinsam verwendeten Schlüssel befinden. Bei einer WAN-Boot-Installation dienen die Schlüssel dem Clientsystem zur Überprüfung der Integrität der vom Server übertragenen Daten und Dateien oder zum Entschlüsseln dieser.

Klonsystem Ein System, das mithilfe eines Solaris Flash-Archivs installiert wurde. Das Klonsystem hat dieselbe Installationskonfiguration wie das Mastersystem.

Kritische Dateisysteme Für das Solaris-BS unabdingbare Dateisysteme. Wenn Sie Solaris Live Upgrade verwenden, sind diese Dateisysteme separate Einhängepunkte in der Datei `vfstab` der aktiven und der inaktiven Boot-Umgebung. Dateisysteme sind beispielsweise `root` (/), `/usr`, `/var` und `/opt`. Diese Dateisysteme werden immer von der Quelle in die inaktive Boot-Umgebung kopiert.

LAN (Local Area Network) Eine Gruppe von nahe beieinander installierten Computersystemen, die über Verbindungshardware und -software miteinander kommunizieren können.

LDAP (Lightweight Directory Access Protocol) Ein erweiterbares Standardprotokoll für den Zugriff auf Verzeichnisse, das bei der Kommunikation zwischen Clients und Servern des LDAP-Namen-Services zum Einsatz kommt.

locate (Gebietsschema) Ein Gebiet in geografischen oder politischen Grenzen, in dem die gleiche Sprache, die gleichen Sitten und die gleichen kulturellen Konventionen gelten. Die englische Sprachumgebung für die USA heißt zum Beispiel `en_US`, die für Großbritannien heißt `en_UK`.

Logisches Gerät Eine Gruppe physischer Speicherbereiche auf einer oder mehreren Festplatten, die im System als ein einziges logisches Gerät erscheinen. In Solaris Volume Manager wird ein logisches Gerät Volume genannt. Für eine Anwendung oder ein Dateisystem sind Volumes, was ihre Funktionsweise angeht, mit einer physischen Festplatte identisch.

Manifest-Teil Ein Teil eines Solaris Flash-Archivs, der zur Überprüfung des Klonsystems dient. Im Manifest-Teil sind die Dateien eines Systems aufgeführt, die auf dem Klonsystem beibehalten, ergänzt oder gelöscht werden sollen. Dieser Teil ist rein informativ. Die Dateien sind in einem internen Format aufgeführt, sodass dieser Teil nicht zum Skripting verwendet werden kann.

Mastersystem Ein System, mit dem ein Solaris Flash-Archiv erstellt wird. Die Systemkonfiguration wird in dem Archiv gespeichert.

MD5 (Message Digest 5) Eine iterative kryptographische Hash-Funktion für die Meldungs-Authentifizierung, einschließlich digitaler Signaturen. Diese Funktion wurde 1991 von Rivest entwickelt.

Medienserver Siehe *Installationsserver*.

`menu.lst` **(Datei)** **Nur x86**: Eine Datei mit einer Liste aller Betriebssysteme, die auf dem betreffenden System installiert sind. Der Inhalt dieser Datei legt fest, welche Betriebssysteme im GRUB-Hauptmenü erscheinen. Über das GRUB-Hauptmenü können Sie komfortabel ein bestimmtes Betriebssystem starten, ohne dafür die `fdisk`-Partitionseinstellungen oder die BIOS-Konfiguration ändern zu müssen.

Metagerät Siehe *Volume*.

Miniroot Ein minimales root-Dateisystem mit Bootfähigkeit (`/`), das auf dem Solaris-Installationsdatenträger enthalten ist. Eine Miniroot besteht aus der Solaris-Software, mit der Systeme installiert und aktualisiert werden können. Auf x86-basierten Systemen wird die Miniroot in das System kopiert, damit es dort als Failsafe-Bootarchiv verfügbar ist. Siehe *Failsafe-Bootarchiv*.

Mirror Siehe *RAID-1-Volume*.

Namen-Server Ein Server, der den Systemen in einem Netzwerk einen Namen-Service zur Verfügung stellt.

Namen-Service Eine verteilte Netzwerkdatenbank, die grundlegende Systeminformationen über alle Systeme im Netzwerk enthält, so dass die Systeme miteinander kommunizieren können. Ist ein Namen-Service vorhanden, können die Systeminformationen netzwerkweit gepflegt und verwaltet und es kann netzwerkweit darauf zugegriffen werden. Ohne Namen-Service muss auf jedem System eine eigene Kopie der Systeminformationen gepflegt werden (in den lokalen `/etc`-Dateien). Sun unterstützt die folgenden Namen-Services: LDAP, NIS und NIS+.

Netzwerkinstallation Eine Möglichkeit, Software über das Netzwerk zu installieren, und zwar von einem System mit CD-ROM- oder DVD-ROM-Laufwerk auf einem System ohne CD-ROM- oder DVD-ROM-Laufwerk. Für Netzwerkinstallationen sind ein *Namen-Server* und ein *Installationsserver* erforderlich.

Nicht-globale Zone Eine virtuelle Betriebssystemumgebung, die in einer bestimmten Instanz des Betriebssystems Solaris erstellt wurde. In einer nicht-globalen Zone können Anwendungen ausgeführt werden, ohne dass sie in irgendeiner Form mit dem Rest des Systems interagieren. Nicht-globale Zonen werden auch kurz als Zonen bezeichnet. Siehe auch *Solaris Zones* und *globale Zone*.

Nicht vernetzte Systeme Systeme, die nicht an ein Netzwerk angeschlossen sind und keine anderen Systeme benötigen.

NIS Der Netzwerkinformationsservice von SunOS 4.0 (Minimum). Eine verteilte Netzwerkdatenbank mit grundlegenden Informationen über die Systeme und die Benutzer im Netzwerk. Die NIS-Datenbank wird auf dem Master-Server und allen Slave-Servern gespeichert.

NIS+ Der Netzwerkinformationsservice von SunOS 5.0 (Minimum). NIS+ ersetzt NIS, den Netzwerkinformationsservice SunOS 4.0 (Minimum).

`/opt`-Dateisystem Ein Dateisystem, das die Einhängepunkte für Software von Drittanbietern und nicht in einem Package enthaltene Software enthält.

Package Eine Sammlung von Software, die für die modulare Installation zu einer Einheit zusammengefasst wurde. Die Solaris-Software ist in mehrere *Softwaregruppen* eingeteilt, die jeweils aus Clustern und *Packages* bestehen.

Patch Analyzer Ein Skript, das Sie von Hand oder als Teil des Solaris-Installationsprogramms ausführen können. Patch Analyzer analysiert das System und ermittelt, welche Patches gegebenenfalls bei einem Upgrade auf ein aktualisiertes Solaris-Release entfernt werden.

Pfeiltasten Die vier Richtungstasten auf dem numerischen Tastenblock.

Plattformgruppe Eine vom Anbieter definierte Gruppe von Hardwareplattformen für die Distribution einer bestimmten Software. Beispiele für gültige Plattformgruppen sind i86pc und sun4u.

Plattformname Die Ausgabe des Befehls `uname -i`. Der Plattformname der Ultra 60 lautet beispielsweise SUNW,Ultra-60.

pool Eine logische Gruppe von Geräten, die das Layout und die physischen Merkmale des verfügbaren ZFS-Speichers beschreibt. Datensätzen wird Speicher aus einem Pool zugewiesen.

Power Management Software, die den Status eines Systems automatisch speichert und dieses System nach 30 Minuten Inaktivität herunterfährt. Wenn Sie die Solaris-Software auf einem System installieren, das der Version 2 der Energy Star-Richtlinien der amerikanischen Umweltbehörde entspricht, wird die Power Management-Software standardmäßig installiert. Ein Beispiel für ein System, bei dem die Power Management-Software standardmäßig installiert ist, ist ein `sun4u` SPARC-basiertes System. Nach einem Neustart werden Sie dann dazu aufgefordert, Power Management zu aktivieren bzw. zu deaktivieren.

Die Energy Star-Richtlinien erfordern, dass Systeme bzw. Monitore automatisch in einen "Sleep-Modus" (Verbrauch von 30 Watt oder weniger) wechseln, in welchem System oder Monitor inaktiv werden.

Primäres Boot-Archiv Ein Boot-Archiv, das zum Booten von Solaris verwendet wird. Dieses Archiv wird auch als „primäres“ Boot-Archiv bezeichnet. Siehe *Boot-Archiv*.

Privater Schlüssel Auch private key. Der Entschlüsselungs-Code für die Verschlüsselung mit öffentlichen Schlüsseln (public-key).

Probe-Schlüsselwort Ein syntaktisches Element, das bei der benutzerdefinierten JumpStart-Installation Attributinformationen über ein System abruft. Im Gegensatz zu Regeln ist es bei Probe-Schlüsselwörtern nicht erforderlich, Übereinstimmungskriterien einzurichten und Profile auszuführen. Siehe auch *Regel*.

Profil Eine Textdatei, in der festgelegt ist, wie die Solaris-Software bei einem benutzerdefinierten JumpStart-Verfahren installiert werden soll. So ist in einem Profil beispielsweise die zu installierende Softwaregruppe definiert. Jede Regel spezifiziert ein Profil, das definiert, wie ein System installiert werden soll, wenn es der Regel entspricht. Sie können für jede Regel ein eigenes Profil erstellen. Sie können ein Profil jedoch auch in mehreren Regeln verwenden. Siehe auch *rules-Datei*.

Profildiskette Eine Diskette mit allen wichtigen, benutzerdefinierten JumpStart-Dateien im Root-Verzeichnis (JumpStart-Verzeichnis).

Profilserver Ein Server mit allen wichtigen, benutzerdefinierten JumpStart-Dateien in einem JumpStart-Verzeichnis.

public key (Öffentlicher Schlüssel) Der Verschlüsselungsschlüssel, der bei der Verschlüsselung mit öffentlichen Schlüsseln zum Einsatz kommt.

public-key cryptography (Verschlüsselung mit öffentlichen Schlüsseln) Ein Kryptographiesystem, bei dem zwei Schlüssel verwendet werden: ein öffentlicher, allen bekannter Schlüssel und ein privater Schlüssel, den nur der Nachrichtenempfänger kennt.

RAID-0-Volume Eine Volumenart, bei der es sich um einen Streifen (Stripe) oder eine Verkettung handeln kann. Diese Komponenten werden auch Submirrors genannt. Ein Stripe oder eine Verkettung stellt den Grundbaustein für einen Mirror dar.

RAID-1-Volume Eine Volume-Art, bei der Daten durch die Vorhaltung mehrerer Kopien repliziert werden. Ein RAID-1-Volume besteht aus einem oder mehreren RAID-0-Volumes; diese werden *Submirrors* genannt. RAID-1-Volumes werden manchmal auch als *Mirrors* bezeichnet.

RAID-Z Storage Pool Ein virtuelles Gerät, das Daten und Parität auf mehreren Platten speichert, die als ein ZFS-Speicherpool verwendet werden können. RAID-Z ähnelt RAID-5.

Regel Eine Folge von Werten, die einem Profil eine oder mehrere Systemattribute zuordnet. Bei benutzerdefinierten JumpStart-Installationen werden Regeln eingesetzt.

Root Als Stamm- oder „Root“-Ebene bezeichnet man die oberste Ebene in einer Elementhierarchie. Alle anderen Elemente sind vom Stamm- bzw. Root-Element abhängig. Siehe *Rootverzeichnis* oder *root* (`/`) *Dateisystem*.

Root-Dateisystem (/) — Das oberste Dateisystem, das alle anderen Dateisysteme unter sich enthält. Alle anderen Dateisysteme sind im Root-Dateisystem (/) eingehängt, und dieses wird niemals ausgehängt. Das Root-Dateisystem (/) enthält die Verzeichnisse und Dateien, die für den Systembetrieb unverzichtbar sind, zum Beispiel den Kernel, die Gerätetreiber und die Programme, die zum Starten (Booten) eines Systems verwendet werden.

Root-Verzeichnis — Die oberste Verzeichnisebene, die alle anderen Verzeichnisse unter sich enthält.

`rules`-Datei — Eine Textdatei, die eine Regel für jede Gruppe von Systemen oder für Einzelsysteme enthält, die automatisch installiert werden sollen. Jede Regel charakterisiert eine Gruppe von Systemen auf der Grundlage von einem oder mehreren Systemattributen. Die Datei `rules` verknüpft jede Gruppe mit einem Profil, einer Textdatei, die definiert, wie die Solaris-Software auf allen Systemen in der Gruppe installiert wird. Eine `rules`-Datei kommt bei benutzerdefinierten JumpStart-Installationen zum Einsatz. Siehe auch *Profil*.

Schlüssel — Der Code zum Ver- oder Entschlüsseln von Daten (auch als „Key“ bezeichnet). Siehe auch encryption (Verschlüsselung).

server — Ein Netzwerkgerät, das Ressourcen verwaltet und einem Client Dienste zur Verfügung stellt.

SHA1 — (Secure Hashing Algorithm) Dieser Algorithmus erzeugt Meldungs-Digests für Eingaben mit einer Länge von weniger als 2^{64}.

snapshot — Ein schreibgeschütztes Abbild eines ZFS-Dateisystems oder eines Volumens einem bestimmten Zeitpunkt.

Softwaregruppe — Eine logische Zusammenstellung der Solaris-Software (bestehend aus Clustern und Packages). Bei einer Solaris-Installation können Sie eine der folgenden Softwaregruppen installieren: die Softwaregruppen für die Hauptsystemunterstützung, Endbenutzer, Entwickler, die gesamte Solaris-Softwaregruppe und, nur auf SPARC-Systemen, die gesamte Solaris-Softwaregruppe plus OEM-Unterstützung.

Softwaregruppe für zentrales System (Core) — Eine Softwaregruppe, die die zum Booten und zum Ausführen des Solaris-BS auf einem System erforderliche Minimalsoftware enthält. Core enthält etwas Netzwerksoftware sowie die Treiber zum Ausführen des CDE-Desktop (Common Desktop Environment-Desktop). Die CDE-Software selbst enthält sie nicht.

Softwaregruppe mit eingeschränkter Netzwerkunterstützung — Eine Softwaregruppe, die den zum Booten und Ausführen eines Solaris-Systems mit eingeschränkter Netzwerkunterstützung mindestens erforderlichen Code enthält. Die Softwaregruppe mit eingeschränkter Netzwerkunterstützung bietet eine textbasierte Mehrbenutzerkonsole und Dienstprogramme für die Systemverwaltung. Mit dieser Softwaregruppe kann ein System Netzwerkschnittstellen erkennen, aktiviert aber keine Netzwerkdienste.

Solaris-DVD- oder -CD-Abbilder — Die Solaris-Software, die auf einem System installiert wird und die auf Solaris-DVDs, -CDs oder der Festplatte eines Installationsservers zur Verfügung steht, auf die die Solaris-DVD- oder -CD-Abbilder kopiert wurden.

Solaris Flash Eine Solaris-Installationsfunktion, mit deren Hilfe Sie ein Archiv der Dateien auf einem System erstellen können *Mastersystem* genannt). Mithilfe dieses Archivs können dann weitere Systeme installiert werden. Diese sind in ihrer Konfiguration mit dem Mastersystem identisch. Siehe auch *Archiv*.

Solaris-Installationsprogramm Ein Installationsprogramm mit einer grafischen Benutzeroberfläche (GUI) oder Befehlszeilenschnittstelle (CLI), das den Benutzer mithilfe von Assistentenfenstern Schritt für Schritt durch die Installation der Solaris-Software und die Software von Drittanbietern führt.

Solaris Live Upgrade Eine Upgrade-Methode, bei welcher das Upgrade in einer zuvor duplizierten Boot-Umgebung ausgeführt wird, während die aktive Boot-Umgebung weiter in Betrieb ist, so dass es nicht zu Ausfällen der Produktionsumgebung kommt.

Solaris-Softwaregruppe für Endbenutzer Eine Softwaregruppe, die die Softwaregruppe für zentrales System (Core) plus die empfohlene Software für einen Endbenutzer enthält, einschließlich Common Desktop Environment (CDE) und DeskSet-Software.

Solaris-Softwaregruppe für Entwickler Eine Softwaregruppe, die die Solaris-Softwaregruppe für Endanwender und zusätzlich die Bibliotheken, Include-Dateien, Manpages und Programmiertools für die Entwicklung von Software enthält.

Solaris Zones Eine Software-Partitionierungstechnologie, die zum Virtualisieren von Betriebssystemdiensten und Bereitstellen einer isolierten, sicheren Umgebung zum Ausführen von Anwendungen dient. Indem Sie eine nicht-globale Zone erstellen, erzeugen Sie eine Umgebung für die Ausführung von Anwendungen, in der Prozesse von allen anderen Zonen isoliert sind. Durch diese Isolierung wird verhindert, dass Prozesse, die in einer Zone laufen, Prozesse in anderen Zonen überwachen oder in sie eingreifen. Siehe auch *Globale Zone* und *Nicht-globale Zone*.

Speicherbereich Auch „Partition“. Die Einheiten, in die der Platz auf der Festplatte von der Software unterteilt wird.

SSL (Secure Sockets Layer) Eine Softwarebibliothek, die eine sichere Verbindung zwischen zwei Seiten (Client und Server) ermöglicht und zur Implementierung von HTTPS, der sicheren Version von HTTP, verwendet wird.

Standalone Ein Computer, der als eigenständiges Gerät läuft und keine Unterstützung durch andere Rechner benötigt.

State Database Eine Statusdatenbank oder State Database speichert Informationen zum Status Ihrer Solaris Volume Manager-Konfiguration auf einer Festplatte ab. Die State Database ist eine Sammlung aus mehreren replizierten Kopien der Datenbank. Jede dieser Kopien wird als *Statusdatenbankreplikation* bezeichnet. Die Statusdatenbank überwacht und speichert Angaben zu Speicherort und Status aller bekannten Statusdatenbankreplikationen.

State Database Replica Eine Kopie einer Statusdatenbank. Die Replica garantiert die Integrität der Datenbankdaten.

Submirror Siehe *RAID-0-Volume*.

Subnetz Ein Schema, bei dem ein logisches Netzwerk in kleinere physische Netzwerke zerlegt wird, um das Routing zu vereinfachen.

Superuser Ein besonderer Benutzer, der berechtigt ist, alle Administrationsvorgänge auf dem System auszuführen. Der Superuser kann lesend und schreibend auf alle Dateien zugreifen, er kann alle Programme ausführen und die Beendigung beliebiger Prozesse erzwingen.

Swap-Bereich Ein Speicherbereich oder eine Datei zur temporären Aufnahme von Hauptspeicherinhalten, bis diese Inhalte wieder in den Hauptspeicher zurückgeladen werden können. Auch bekannt als Dateisystem `/swap` oder `swap`.

`sysidcfg` Eine Datei, in der eine Reihe spezieller Systemkonfigurationsschlüsselwörter angegeben werden können, die ein System vorkonfigurieren.

Systemkonfigurations- datei (`system.conf`) Eine Textdatei, in der Sie angeben, wo die Datei `sysidcfg` und die Dateien für die benutzerdefinierte JumpStart-Installation gespeichert sind, die Sie für eine WAN-Boot-Installation verwenden möchten.

Teilnetzmaske Eine Bit-Maske zur Auswahl von Bits aus einer IP-Adresse für die Adressierung eines Teilnetzes. Die Maske ist 32 Bit lang und wählt den Netzwerkanteil der IP-Adresse sowie 1 oder mehrere Bits des lokalen Adressanteils aus.

time zone (Zeitzone) Die 24 nach Längengraden eingeteilten Abschnitte der Erdoberfläche, für die eine bestimmte Standardzeit gilt.

`truststore`**-Datei** Eine Datei, die ein oder mehrere digitale Zertifikate enthält. Bei einer WAN-Boot-Installation überprüft das Clientsystem auf Grundlage der Daten in der Datei `truststore` die Identität des Servers, der die Installation durchzuführen versucht.

Upgrade, Aufstufung, Aufrüstung Eine Installation, bei der neue Dateien mit vorhandenen vereint und Änderungen soweit wie möglich beibehalten werden.

Ein Upgrade des Solaris-BS vereint die neue Solaris-Version mit den auf der Systemfestplatte (bzw. Festplatten) vorhandenen Dateien. Dabei werden möglichst viele der Änderungen gespeichert, die Sie an der vorherigen Version des Solaris-BS vorgenommen haben.

Upgrade-Option Eine Option des Programms Solaris-Installationsprogramm . Bei einem Upgrade wird die neue Version von Solaris mit den vorhandenen Dateien auf der Festplatte bzw. den Festplatten zusammengeführt. Bei einem Upgrade werden möglichst viele der lokalen Modifikationen beibehalten, die seit der letzten Installation von Solaris vorgenommen wurden.

URL (Uniform Resource Locator) Das Adressiersystem, mit dessen Hilfe Client und Server Dokumente abrufen. Ein URL wird auch häufig als Position bezeichnet. URLs haben das Format *Protokoll://Rechner:Port/Dokument*.

Ein Beispiel ist `http://www.Beispiel.com/index.html`.

/usr-Dateisystem Ein Dateisystem auf einem Standalone-System oder Server, das viele der Standard-UNIX-Programme enthält. Die gemeinsame Nutzung des großen Dateisystems `/usr` auf einem Server statt der Pflege einer lokalen Kopie dieses Dateisystems verringert den Gesamtbedarf an Festplattenplatz zum Installieren und Ausführen der Solaris-Software auf einem System.

/var-Dateisystem Ein Dateisystem oder Verzeichnis (auf Standalone-Systemen) mit Systemdateien, die sich im Zuge der Systemnutzung in der Regel ändern oder wachsen. Zu diesen Dateien gehören Systemprotokolle, `vi`-Dateien, Mail-Dateien und UUCP-Dateien.

Verkettung Ein RAID-0-Volume. Bei der Verkettung von Speicherbereichen werden Daten so lange auf den·ersten verfügbaren Speicherbereich geschrieben, bis dieser voll ist. Sobald ein Speicherbereich voll ist, werden die Daten auf den·jeweils folgenden Speicherbereich geschrieben. Verkettungen bieten keine Datenredundanz, es sei denn, sie sind Bestandteil eines Mirrors. Siehe auch: RAID-0-Volume.

Vernetzte Systeme Eine Gruppe von Systemen („Hosts“ genannt), die über Hardware und Software verbunden sind, so dass sie miteinander kommunizieren und Informationen austauschen können. Ein solches System wird als Local Area Network (lokales Netzwerk - LAN) bezeichnet. In vernetzten Systemen sind in der Regel ein oder mehrere Server erforderlich.

Virtuelles Gerät Ein logisches Gerät in einem ZFS-Pool, bei dem es sich um eine physikalisches Gerät, eine Datei oder eine Sammlung von Geräten handeln kann.

volume Eine Gruppe physischer Speicherbereiche oder anderer Volumes, die im System als ein einziges logisches Gerät erscheinen. Für eine Anwendung oder ein Dateisystem sind Volumes, was ihre Funktionsweise angeht, mit einer physischen Festplatte identisch.

In manchen Befehlszeilen-Dienstprogrammen werden Volumes auch Metageräte genannt. Für Volumes werden auch die Standard-UNIX-Begriffe *Pseudogerät* oder *virtuelles Gerät* verwendet.

Volume Manager Ein Programm, das einen Mechanismus zum Verwalten und Zugreifen auf die Daten auf DVD-ROMs, CD-ROMs und Disketten zur Verfügung stellt.

WAN (Wide Area Network) Ein Netzwerk, das mehrere LANs (Local Area Networks) oder Systeme an verschiedenen geografischen Standorten über Telefon-, Glasfaserleitung oder Satellit miteinander verbindet.

WAN-Boot-Installation Eine Installationsart, die es ermöglicht, Software mithilfe von HTTP oder HTTPS über ein WAN (Wide Area Network) zu booten und zu installieren. Mit dem WAN-Boot-Installationsverfahren können Sie ein verschlüsseltes Solaris Flash-Archiv über ein öffentliches Netzwerk senden und auf einem Remote-Client eine benutzerdefinierte JumpStart-Installation durchführen.

WAN-Boot-Miniroot Eine Miniroot, die im Hinblick auf die Durchführung einer WAN-Boot-Installation verändert wurde. Die WAN-Boot-Miniroot enthält einen Teilsatz der Software in der Solaris-Miniroot. Siehe auch Miniroot.

WAN-Boot-Server Ein Webserver, der die für eine WAN-Boot-Installation benötigten Konfigurations- und Sicherheitsdateien bereitstellt.

wanboot-cgi-**Programm**	Das CGI-Programm, das die für eine WAN-Boot-Installation benötigten Daten und Dateien abruft und überträgt.
wanboot.conf-**Datei**	Eine Textdatei, in der Sie die Konfigurationsinformationen und Sicherheitseinstellungen angeben, die für die Durchführung einer WAN-Boot-Installation benötigt werden.
wanboot-**Programm**	Das sekundäre Boot-Programm, das die WAN-Boot-Miniroot, die Client-Konfigurationsdateien und die für eine WAN-Boot-Installation erforderlichen Installationsdateien lädt. Bei WAN-Boot-Installationen führt das Binärprogramm wanboot ähnliche Vorgänge wie die sekundären Boot-Programme ufsboot oder inetboot durch.
ZFS	Ein Dateisystem, das Speicherpools zum Verwalten des physischen Speichers verwendet.
Zone	Siehe *Nicht-globale Zone*

Index

Zahlen und Symbole

#
  in Profilen, 39
  in `rules`-Dateien, 35
&& (Ampersands) Regelfeld, 36
(/) Dateisysteme
  von JumpStart gesetzter Wert, 165
! (Ausrufezeichen) rule-Feld, 35
= (Gleichheitszeichen) in Profilfeld, 61

A

Abgeleitete Profile, 60
abgeleitete Profile, 61
Abgleich, abgeleitete Profile, 61
Abgleichen
  Reihenfolge für Regeln, 37, 86, 93
add_install_client-Befehl,
  JumpStart-Verzeichniszugriff, 28
Alternative Installationsprogramme, 76
Ampersands (&&) Regelfeld, 36
AND-Regelfeld, 36
any
  Probe-Schlüsselwort, Beschreibung und Werte, 169
  Rule-Schlüsselwort, Beschreibung und Werte, 111, 169
Anzeige
  Voraussetzungen für tip-Verbindung, 89, 94
Arbeitsspeicher, und Größe des
  Swap-Speicherplatzes, 130
arch, Probe-Schlüsselwort, 169
arch, Rule-Schlüsselwort, 111, 169
Archiv
  JumpStart-Profilbeispiel, 44, 45, 46, 47
  Schlüsselwörter, benutzerdefiniertes JumpStart, 118-124
archive_location-Schlüsselwort, 118-124
Ausgabedateien
  Finish-Skript-Protokoll, 62
  Protokoll für Begin-Skript, 60
Ausrufezeichen (!) rule-Feld, 35
auto_install_sample-Verzeichnis
  check-Skript, 58, 81
  Dateien in JumpStart-Verzeichnis kopieren, 27, 32, 33
  set_root_pw-Finish-Skript, 66, 67

B

-b, Option des setup_install_server-Befehls, 105
Backslash in `rules`-Dateien, 35
backup_media-Schlüsselwort, 124-126
`begin.log`-Datei, 60
Begin-Regelfeld, Beschreibung, 36
Begin-Skripten
  abgeleitete Profile erstellen, 60, 61
  Berechtigungen, 60
  Regelfeld, 36
  standortspezifische Installationsprogramme, 76
  Überblick, 59
Benennung, `custom_probes`-Datei, 78
Benutzerdefinierte JumpStart-Installation, 83

Benutzerdefinierte JumpStart-Installation *(Fortsetzung)*
Beispiele, 101, 110
Booten und Installieren, 110
check-Skript, 108
eng_profile erstellen, 106
Entwicklungssysteme einrichten, 108
JumpStart-Verzeichnis, 105
marketing_profile erstellen, 106
Marketing-Systeme einrichten, 104, 109
Netzwerk, 20
nicht vernetzte Systeme, 19
Profile für RAID-1-Volumes, 48, 50
rules-Datei bearbeiten, 107
Solaris Flash-Profil, 44, 45, 46, 47
Standalone-Systeme, 19
Standortkonfiguration, 102
WAN-Boot-Installationsprofil, 45
Beschreibung, 21
Booten und Installieren, 83
optionale Funktionen, 59
Begin-Skripten, 59, 61
Finish-Skripten, 62, 67
standortspezifische Installationsprogramme, 76
Überblick, 59
Profilschlüsselwörter, 116
Überblick, 21
Voraussetzungen für tip-Verbindung, 89, 94
vorbereiten, 21, 58
Berechtigungen
Begin-Skripten, 60
Finish-Skripten, 62
JumpStart-Verzeichnis, 25, 30
boot: cannot open /kernel/unix, Meldung, 186
boot_device-Schlüsselwort, 126
Booten
mit GRUB, Befehlsreferenz, 98
mit GRUB installieren, 94, 96
Profildiskette erstellen, 32
bootenv createbe-Schlüsselwort, 128
bootparams, Datei, aktualisieren, 191
bootparams-Datei, JumpStart-Verzeichniszugriff gewähren, 29
Bourne-Shell-Skripten in Regelfeldern, 36

C

-c, Option
`pfinstall`-Befehl, 56
add_install_client-Befehl, 108, 109
Can't boot from file/device, Meldung, 186
check-Skript
abgeleitete Profile und, 61
`custom_probes`-Datei validieren, 82
`custom_probes.ok`-Datei erstellen, 81
Datei `custom_probes` validieren, 81
Regeln testen, 58, 82
`rules`-Datei validieren, 58, 82
Check-Skript, `Rules`-Datei Validierung, 57
check-Skript
`rules.ok`-Datei erstellen, 57
client_arch-Schlüsselwort, 129
CLIENT MAC ADDR, Fehlermeldung, 191
client_root profile-Schlüsselwort, 130
clock gained xxx days, Meldung, 186
cluster-Profilschlüsselwort
Beispiele, 40
Beschreibung und Werte, 131-132, 132
CPUs (Prozessoren)
Probe-Schlüsselwörter, 169
Rule-Schlüsselwörter, 111, 169
`.cshrc`-Datei, 66
`custom_probes`-Datei
benennen, 78
`custom_probes`, 82
mit check-Skript validieren, 81, 82
Voraussetzungen, 78
`custom_probes.ok`-Datei
Beschreibung, 81
erstellen, 81
Erstellen, 81

D

Dateien oder Dateisysteme
Ausgabe von Begin-Skript, 60
Finish-Skript-Ausgabe, 62
kopieren
JumpStart-Verzeichnisdateien mittels Finish-Skripten, 63

Dateien und Dateisysteme
erstellen
lokale Dateisysteme, 138-141
RAID-1-Volumes, 141-142
Erstellen eines UFS, 31
kopieren
JumpStart-Installationsdateien, 27, 32, 33
Remote-Dateisysteme einhängen, 137-138
dfstab-Datei, 26, 105
Disketten
JumpStart-Verzeichniszugriff, 28
x86: JumpStart-Verzeichnis, 30
Diskless-Clients
Plattformen, 129
Swap-Speicherplatz, 130
disks, Probe-Schlüsselwort, Beschreibung und Werte, 169
disksize, Rule-Schlüsselwort, Beschreibung und Werte, 112, 169
domainname, Probe-Schlüsselwort, 169
domainname, Rule-Schlüsselwort, 112, 169
Domänen
Probe-Schlüsselwort, 169
Rule-Schlüsselwort, 112, 169
dontuse-Profilschlüsselwort, 133-134, 166

E

Einhängen
durch Solaris-Installation, 62
Remote-Dateisysteme, 137-138
Vorsichtsmaßnahmen bei Begin-Skripten, 60
Einschränkungen für ZFS, 172
eng_profile, Beispiel, 106
Erstellen
`custom_probes.ok`-Datei, 81
Festplattenkonfigurationsdateien, 70
JumpStart-Verzeichnis, auf dem Server, 25
lokale Dateisysteme, 138-141
Profile
abgeleitet, 60
Beschreibung, 38
RAID-1-Volumes, 141-142
rules-Datei, 34
Erstellen *(Fortsetzung)*
`rules.ok`-Datei, 57, 81
UFS, 31
`/etc/bootparams`, Datei, JumpStart-Verzeichniszugriff aktivieren, 191
`/etc/bootparams`-Datei, JumpStart-Verzeichniszugriff gewähren, 29
`/etc/dfs/dfstab`-Datei, 26, 105
`/etc/mnttab`-Datei, 31

F

`fdisk`-Befehl, 72
`fdisk`-Profilschlüsselwort
Beispiel, 40
Beschreibung und Werte, 134-136
Fehlerbehebung
allgemeine Installationsprobleme
Booten des Systems, 191
Booten über das Netzwerk mit DHCP, 191
Booten über das Netzwerk mit DHCP, 191
Booten vom falschen Server, 191
Festplatten
einhängen, 137-138
Größe
Probe-Schlüsselwörter, 169, 170
Root-Speicherplatz, 130
Rule-Schlüsselwörter, 112, 116, 169, 170
Partitionierung
Beispiele, 40
für Standardpartitionierung auswählen, 166
Profilschlüsselwort, 157
von Standardpartitionierung ausschließen, 133-134
rootdisk-Werte, 165
Swap-Bereich
Profil-Beispiele, 21
Swap-Speicherplatz
Diskless-Client, 130
Maximalgröße, 130
Profilbeispiele, 40
Festplattenkonfigurationsdateien
Beschreibung, 53, 70

Festplattenkonfigurationsdateien *(Fortsetzung)*
erstellen
SPARC-basierte Systeme, 70
x86-basierte Systeme, 72
filesys-Profilschlüsselwort
Beispiele, 40
Beschreibung und Werte, 137-138
filesys-Schlüsselwort, 138-141, 141-142
`finish.log`-Datei, 62
Finish-Regelfeld, Beschreibung, 36
Finish-Skripten
Packages und Patches hinzufügen, 63
Regelfeld, 36
Root-Passwort eines Systems setzen, 66
Root-Umgebung anpassen, 66

G

geo-Schlüsselwort, 143
Gesamte Solaris-Softwaregruppe, 131-132
Gesamte Solaris-Softwaregruppe plus OEM-Unterstützung, 131-132
getfile: RPC failed: error 5: RPC Timed out, Meldung, 30
Gleichheitszeichen (=) in Profilfeld, 61
Größe
Arbeitsspeicher, 114, 169
Fenstergröße für tip-Verbindung, 89, 94
Festplatte
Probe-Schlüsselwörter, 169, 170
Root-Speicherplatz, 130
Rule-Schlüsselwörter, 112, 116, 169, 170
Swap-Bereich
Profil-Beispiele, 21
Swap-Speicherplatz
Diskless-Client, 130
Maximalgröße, 130
GRUB-basiertes Booten
Befehlsreferenz, 98
installieren, 94, 96
Profildiskette erstellen, 32

H

Hinzufügen
Cluster bei einem Upgrade, 133
Packages und Patches mittels Finish-Skript, 63
Pakages aus Softwaregruppen, 151
Regeln zur rules-Datei, 35
hostaddress, Probe-Schlüsselwort, 169
hostaddress, Rule-Schlüsselwort, 112, 169
hostname, Probe-Schlüsselwort, Beschreibung und Werte, 169
hostname, Rule-Schlüsselwort
Beispiel, 111-116
Beschreibung und Werte, 112, 169

I

install_config-Befehl, 29, 30
install_type-Profilschlüsselwort
Beispiele, 40
Profile testen, 56-57
Voraussetzung, 39, 40
install_type-Schlüsselwort, 145
Installation vorbereiten, mit benutzerdefiniertem JumpStart, 21
Installationsvorbereitung, bei benutzerdefiniertem JumpStart, 58
installed, Probe-Schlüsselwort, Beschreibung und Werte, 169
installed, Rule-Schlüsselwort, Beschreibung und Werte, 113, 169
IP-Adressen
Probe-Schlüsselwort, 169
Rule-Schlüsselwort, 112, 169

J

JumpStart-Verzeichnis
Beispiel einer rules-Datei, 35
Berechtigungen, 25, 30
Dateien kopieren
Installationsdateien, 27, 32, 33
mittels Finish-Skripten, 63
Dateien mittels Finish-Skripten hinzufügen, 63

JumpStart-Verzeichnis *(Fortsetzung)*
erstellen
Beispiel, 105
Diskette für SPARC-basierte Systeme, 30
Diskette für x86-basierte Systeme, 30, 33
Server, 25
gemeinsam nutzen, 25, 105
JumpStart-Verzeichnis freigeben, 26
JumpStart-Verzeichnis gemeinsam nutzen, 105

K

karch, Probe-Schlüsselwort, 169
karch, Rule-Schlüsselwort, 113, 169
Kommentare
in Profilen, 39
in `rules`-Dateien, 35
Konfigurieren, Festplattenkonfigurationsdateien erstellen, 70
Kopieren
JumpStart-Installationsdateien, 27, 32, 33
JumpStart-Verzeichnisdateien, 63

L

layout_constraint-Schlüsselwort, 145-148
le0: No carrier - transceiver cable problem, Meldung, 186
locale-Schlüsselwort, 149
logisches UND-Regelfeld, 36
Löschen, Cluster bei einem Upgrade, 133

M

marketing_profile, Beispiel, 106
Mehrere Zeilen in `rules`-Dateien, 35
memsize, Probe-Schlüsselwort, Beschreibung und Werte, 169
memsize, Rule-Schlüsselwort, Beschreibung und Werte, 114, 169
`metadb`-Profilschlüsselwort, 149
Mikroprozessoren
Probe-Schlüsselwörter, 169
Rule-Schlüsselwörter, 111, 169
mnttab-Datei, 31
model, Probe-Schlüsselwort, Beschreibung und Werte, 169
model, Rule-Schlüsselwort, Beschreibung und Werte, 114, 169

N

Namen/Benennung
abgeleitete Profilnamen, 61
Host-Name, 112, 169
rules-Datei, 35
System-Modellnamen, 114, 169
network, Probe-Schlüsselwort, Beschreibung und Werte, 169
network, Rule-Schlüsselwort, Beschreibung und Werte, 114, 169
Netzwerkinstallation, Benutzerdefinierte JumpStart-Installation, Beispiel, 20
Netzwerknummer, 114, 169
No carrier - transceiver cable problem, Meldung, 186
no_master_check-Schlüsselwort, 150
noneuclidean-Profilschlüsselwort, 150
Not a UFS filesystem, Meldung message, 186

O

osname, Probe-Schlüsselwort, 169
osname, Rule-Schlüsselwort, 115, 169

P

-p, Option des check-Skript, 58
-p, Option von check-Skript, 81
Packages
Administrationsdatei, 59
Anforderungen bei der Verwendung der benutzerdefinierten JumpStart-Installation, 205

Packages *(Fortsetzung)*
hinzufügen
mittels chroot, 66
mittels Finish-Skript, 63
Solaris Live Upgrade
Anforderungen, 205
Partitionierung
Beispiele, 40
`fdisk`-Partitionen, 40, 134-136
Festplatten ausschließen, 133-134
Profilschlüsselwort, 166
partitioning, Profilschlüsselwort, 157
partitioning-Schlüsselwort, 157
Passwort, Root, 66, 67
Patches
hinzufügen
mittels chroot, 66
mittels Finish-Skript, 63
Pfade, check-Skript, 58, 81
pfinstall-Befehl, 52
Plattformen
Diskless-Client, 129
Probe-Schlüsselwörter, 169
Rule-Schlüsselwörter, 113, 169
System-Modellnamen, 114, 169
Systemattribute und Profile abgleichen, 37, 86, 93
probe, Rule-Schlüsselwort, Beschreibung und Werte, 115
Probe-Schlüsselwörter
arch, 169
disks, 169
domainname, 169
hostaddress, 169
hostname, 169
installed, 169
karch, 169
memsize, 169
model, 169
network, 169
`osname`, 169
rootdisk, 169
totaldisk, 170
Profile
abgeleitete Profile, 60, 61
Profile *(Fortsetzung)*
Beispiele, 40
eng_profile, 106
marketing_profile, 106
Solaris Flash, 44, 45, 46, 47
WAN-Boot-Installation, 45
ZFS, 173
benennen, 39
Beschreibung, 38
erstellen, 38
Kommentare in, 39
mit Systemen abgleichen, 37, 86, 93
Regelfeld, 36
testen, 56-57
Voraussetzungen, 35, 39
Profilschlüsselwörter, 116, 166
archive_location, 118-124
backup_media, 124-126
boot_device, 126
bootenv createbe, 128
bootenv installbe für ZFS, 178
client_arch, 129
client_root, 130
client_swap, 130
cluster
Beispiele, 40
Beschreibung und Werte, 131-132, 132
dontuse
Beschreibung und Werte, 133-134
usedisk und, 166
`fdisk`
Beispiel, 40
Beschreibung und Werte, 134-136
filesys
Beispiele, 40
Beschreibung und Werte, 137-138
lokale Dateisysteme, 138-141
RAID-1-Volumes, 141-142
Remote-Dateisysteme, 137-138
forced_deployment, Beschreibung und Werte, 143
geo
Beschreibung und Werte, 143
Groß- und Kleinschreibung, 116

Profilschlüsselwörter *(Fortsetzung)*
install_type
Beispiele, 40
Beschreibung und Werte, 145
für ZFS, 179
Voraussetzung, 39, 40
layout_constraint, Beschreibung und Werte, 145-148
local_customization, Beschreibung und Werte, 148
locale, Beschreibung und Werte, 149
metadb
Beispiele, 40
Beschreibung und Werte, 149
no_master_check, Beschreibung und Werte, 150
noneuclidean, 150
Partitionierung
Beispiele, 40
Festplatten ausschließen, 133-134
Festplatten auswählen, 166
partitioning
Beschreibung und Werte, 157
Pool für ZFS, 179
root_device, 164
root_device für ZFS, 181
Schnellreferenz, 116
Statusdatenbankreplikationen erstellen (meatball), 149
system_type
Beispiele, 40
Beschreibung und Werte, 165
usedisk, Beschreibung und Werte, 166
Protokolldateien
Ausgabe von Begin-Skript, 60
Finish-Skript-Ausgabe, 62
Prozessoren
Probe-Schlüsselwörter, 169
Rule-Schlüsselwörter, 111, 169
prtvtoc-Befehl
SPARC: Erstellen einer Festplattenkonfigurationsdatei, 70
x86: Festplattenkonfigurationsdatei erstellen, 73

R

-r, Option des check-Skripts, 58
-r, Option von check-Skript, 82
Regeln
abgeleitete Profile, 60
Abgleichsreihenfolge, 86, 93
Beispiele, 37
Feldbeschreibungen, 37
Gültigkeit prüfen, 58, 82
mehrzeilige Regeln, 35
Reihenfolge abgleichen, 37
rootdisk matching rules, 165
Übereinstimmungsregeln für rootdisk, 165
Remote-Dateisysteme, einhängen, 137-138
Root (/)-Dateisysteme, Profil-Beispiel, 21
Root-Dateisysteme (/), Package-Anforderungen für eine inaktive Boot-Umgebung, 205
root_device-Schlüsselwort, 164
Root-Passwort, mit einem Finish-Skript setzen, 66
Root-Umgebung, mit Finish-Skript anpassen, 66
rootdisk
Definition, 165
slice-Wert für filesys, 138
von JumpStart gesetzter Wert, 165
RPC failed: error 5: RPC Timed out, Meldung, 30
RPC-Fehler mit Zeitüberschreitung, 191
RPC Timed out, Meldung, 30, 191
rule_keyword-Regelfeld, 36
Rule-Schlüsselwörter, 111
any, Beschreibung und Werte, 111, 169
arch, 111, 169
disksize, Beschreibung und Werte, 112, 169
domainname, 112, 169
hostaddress, 112, 169
rule-Schlüsselwörter, hostname, 111-116
Rule-Schlüsselwörter
hostname, 112, 169
installed, Beschreibung und Werte, 113, 169
karch, 113, 169
memsize, 114, 169
model, 114, 169
network, 114, 169
`osname`, 115, 169
`probe`, 115

Rule-Schlüsselwörter *(Fortsetzung)*
totaldisk, 116, 170
rule_value-Regelfeld, 36
rules
abgeleitete Profile, 61
Feldbeschreibungen, 35
Syntax, 35
rules-Datei
Beispiel, 35
benennen, 35
benutzerdefinierte JumpStart-Installation, Beispiel, 107
Beschreibung, 34
erstellen, 34
Kommentare in, 35
mehrzeilige Regeln, 35
mit check-Skript validieren
abgeleitete Profile und, 61
benutzerdefinierte JumpStart-Installation, Beispiel, 108
mittels check validieren, 58
Regeln hinzufügen, 35
Regeln testen, 58
Syntax, 35
rules.ok-Datei
Abgleichsreihenfolge für Regeln, 86, 93
`rules.ok`-Datei
Beschreibung, 57
erstellen, 57
`rules.ok`-Datei, Reihenfolge für Regeln abgleichen, 37

S

-s, Option des Befehls add_install_client, 109
Schlüsselwörter
Probe, 77
Solaris Flash-Archive, benutzerdefiniertes JumpStart, 118-124
Server
JumpStart-Verzeichnis erstellen, 25
Root-Speicherplatz, 130
set_root_pw -Finish-Skript, 67
set_root_pw-Finish-Skript, 66
share-Befehl
JumpStart-Verzeichnis freigeben, 26
JumpStart-Verzeichnis gemeinsam nutzen, 105
shareall, Befehl, 105
shareall-Befehl, 26
`SI_PROFILE`, Umgebungsvariable, 61
Sicherheit, Root-Passwort, 66, 67
Skripten
Begin-Skripten, 59, 61, 76
Bourne-Shell-Skripten in Regelfeldern, 36
Finish-Skripten, 62, 67, 76
Slices
Profilbeispiele, 40
Softwaregruppe mit eingeschränkter Netzwerkunterstützung, 131-132
Softwaregruppen
aktualisieren, 133
für Profile, 131-132
Profilbeispiele, 40
Solaris-Software
Gruppen, 131-132
aktualisieren, 133
Profilbeispiele, 40
Release oder Version
installed, Probe-Schlüsselwort, 169
installed, Rule-Schlüsselwort, 113, 169
osname, Probe-Schlüsselwort, 169
osname, Rule-Schlüsselwort, 115, 169
Solaris-Softwaregruppe für Endbenutzer, 131-132
Solaris-Softwaregruppe für Entwickler, 131-132
Profilbeispiel, 40
Solaris-Softwaregruppe zur Hauptsystemunterstützung, 131-132
Solaris-Softwareversion
installed, Probe-Schlüsselwort, 169
installed, Rule-Schlüsselwort, 113, 169
`osname`, Probe-Schlüsselwort, 169
`osname`, Rule-Schlüsselwort, 115, 169
Speicher
Probe-Schlüsselwort, 169
Rule-Schlüsselwort, 114, 169
Speicherbereiche
Probe-Schlüsselwort, 169
Rule-Schlüsselwort, 113, 169

Standalone-Systeme
  Beispiel für benutzerdefinierte JumpStart-Installation, 19
  Profilbeispiele, 40
STANDARD-BOOT-GERÄT WECHSELN, Meldung, 192
standortspezifische Installationsprogramme, 76
Starten, check-Skript, 58
stty-Befehl, 89, 94
SUNWCall-Gruppe, 131-132
SUNWCprog-Gruppe, 131-132
SUNWCreq-Gruppe, 131-132
SUNWCrnet-Gruppe, 131-132
SUNWCuser-Gruppe, 131-132
SUNWCXall-Gruppe, 131-132
Swap-Dateisysteme
  Größe ermitteln, 130
  Profil-Beispiele, 21
  Swap-Speicherplatz für Diskless-Clients, 130
  und Speichergröße, 130
system_type-Profilschlüsselwort
  Beispiele, 40
  Beschreibung und Werte, 165

T

Testen
  custom_probes-Dateien , 82
  custom_probes-Dateien validieren
    mit check-Skript, 81
  Profile, 53, 56-57
  rules-Dateien validieren
    abgeleitete Profile und, 61
    benutzerdefinierte JumpStart-Installation, Beispiel, 108
    mit check-Skript, 82
    mittels check, 58
    Regeln testen, 58
  Validieren Rules-Dateien
    Check verwenden, 57
Token-Ring-Karte, Fehler beim Booten, 190
totaldisk, Probe-Schlüsselwort, 170
totaldisk, Rule-Schlüsselwort, 116, 170
transceiver cable problem, Meldung, 186

U

Übereinstimmung
  rootdisk-Werte, 165
UFS, 31
Unbekannter Client (Fehlermeldung), 185
Upgrade
  benutzerdefinierte JumpStart-Installation, 83
  Profilschlüsselwörter, 133, 145, 157
  Upgrade fehlgeschlagen, 197
Upgrade fehlgeschlagen, Probleme beim Neustart, 197
usedisk-Profilschlüsselwort, Beschreibung und Werte, 166

V

Validieren
  custom_probes-Datei
    mit check-Skript, 82
  rules-Dateien
    abgeleitete Profile und, 61
    benutzerdefinierte JumpStart-Installation, Beispiel, 108
  Rules-Dateien
    Check verwenden, 57
  rules-Dateien
    mit check-Skript, 82
    mittels check, 58
    Regeln testen, 58
validieren
  Validieren custom_probes-Datei
    testen, 82
/var/sadm/system/logs/begin.log-Datei, 60
/var/sadm/system/logs/finish.log-Datei, 62
Variablen
  SI_PROFILE, 61
  SYS_MEMSIZE, 55
Version der Solaris-Software
  installed, Probe-Schlüsselwort, 169
  installed, Rule-Schlüsselwort, 113, 169
  osname, Probe-Schlüsselwort, 169

Version der Solaris-Software *(Fortsetzung)*
osname Rule-Schlüsselwort, 115, 169
Verzeichnisse
JumpStart
Beispiel einer rules-Datei, 35
Berechtigungen, 25, 30
Dateien hinzufügen, 63
Dateien kopieren, 63
für Systeme erstellen, 30
Installationsdateien kopieren, 27, 32, 33
Verzeichnis erstellen, 105
Verzeichnis freigeben, 26
Verzeichnis gemeinsam nutzen, 105
wechseln
zu JumpStart-Verzeichnis, 58, 81
zu Solaris-Softwareabbild auf lokalem Datenträger, 27
zu Solaris *SPARC*-Softwareabbild auf lokaler Festplatte, 31
zu Solaris x86-Softwareabbild auf lokaler Festplatte, 33
Verzeichniswechsel
Solaris *SPARC*-Softwareabbild auf lokaler Festplatte, 31
zu JumpStart-Verzeichnis, 58, 81
zu Solaris-Softwareabbild auf lokalem Datenträger, 27
zu Solaris x86-Softwareabbild auf lokaler Festplatte, 33
volcheck-Befehl, 31, 33
Voraussetzungen
custom_probes-Datei, 78
Profile, 35, 39
Voraussetzungen für tip-Verbindung, 89, 94
Vorgaben
abgeleiteter Profilname, 61
installierte Softwaregruppe, 132
Partitionierung
Festplatten ausschließen, 133-134
Festplatten auswählen, 166

W

WARNING: clock gained xxx days, Meldung, 186
WARNUNG: BOOT-GERÄT WECHSELN, 192

Z

Zeilenumbruch in rules-Dateien, 35
ZFS
Einschränkungen, 172
Profilbeispiele, 173
Profilschlüsselwörter
Schnellreferenz, 116
Schlüsselwörter, Beschreibung, 177
Übersicht und Planung, 171